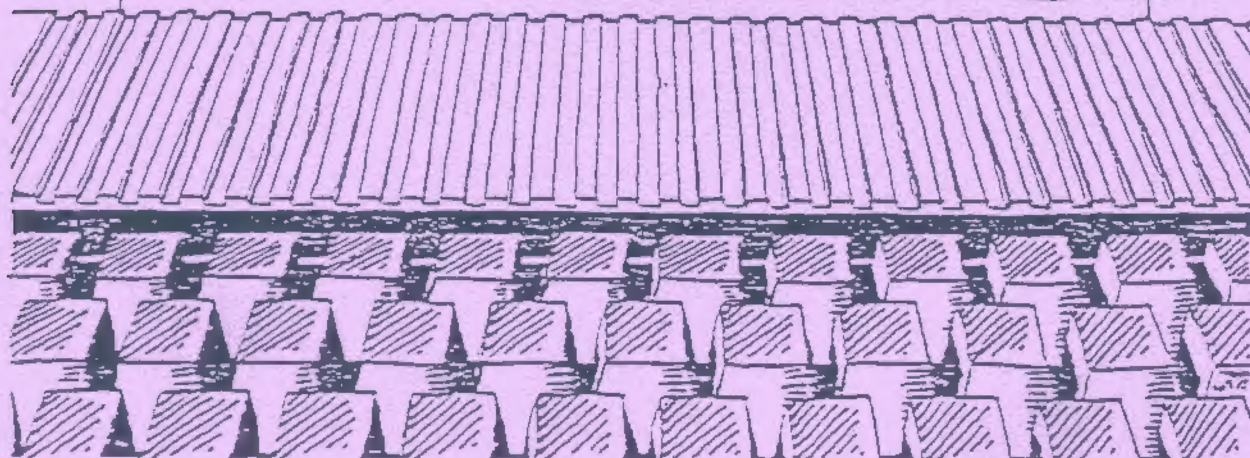
ATOM NIEUWS

JAARGANG : 9
NUMMER : 3
DISKNR. : 5 1990



Bestuur:

Voorzitter:	Secretaris:	Penningmeester:
P. van Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JP Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
Tel.02997-1902	Tel.08373-13757	Tel.03461-3495

Clubwinkel TIJDELIJK!:	Redactie Atom Nieuws:
J.Hartog	E.Sanders
Keyenbergseweg 60	Rosslag 13
6871 WK Renkum	6049 BE Herten
Tel. 08373-13757	Tel. 04750-30401

Contributie 1990 : fl. 25.00 ; Atom Computer Club : Giro 5244293

Redactie Atom Nieuws		SPS-Printservice	Ledenadministratie
E.Sanders	04750-30401	E.Sanders	S.van Leeuwen
B.Tossaint	043-431675	Rosslag 13	Kompasstraat 32
W.Truyen	09-3211564792	6049 BE Herten	1973 PX IJmuiden
R.Leurs		04750-30401	Tel. 02550-22435

Uiterste datum inlevering KOPIJ: nr. 4-9 01 Nov 1990

De clubwinkel:

80-koloms videokaart		fl. 40.00
Geheugenkaart 16 kByte , exclusief onderdelen		fl. 1.00
(NEE, die fl. 1,-- is geen TYPfout!!!)		
Schakelkaart: deze is uitverkocht		
Minischakelkaart: ook deze kaart is uitverkocht		
Combikaart: Ze SPS Print-Service		
Z-80 kaart voor CP/M ,exclusief onderdelen		fl. 50.00
ACORN NIEUWS 1982 , 97 pagina's samenvatting		fl. 2.50
ATOM NIEUWS jaargang 1983, ruim 450 pagina's		fl. 2.50
ATOM NIEUWS jaargang 1984, ruim 650 pagina's		fl. 2.50
ATOM NIEUWS jaargang 1985, ruim 650 pagina's		fl. 2.50
ATOM NIEUWS Jaargang 1986, ruim 500 pagina's		fl. 2.50
ATOM NIEUWS Jaargang 1987, ruim 300 pagina's		fl. 2.50
ATOM-WARE 1 : ATOM Basic interpreter ,	98 pagina's	fl. 1.00
ATOM-WARE 2 : ATOM Disc Operating System ,	68 pagina's	fl. 1.00
ATOM-WARE 3 : ATOM Monitor Operating System ,	80 pagina's	fl. 1.00

LEVERING geschiedt via uw regionale penningmeester , of rechtstreeks via de federatieve penningmeester . Bij rechtstreekse bestelling stort u het bedrag van het gewenste artikel , vermeerderd met fl.4.00 portokosten , op de giro van de federatie onder vermelding van de naam van het artikel en uw lidmaatschapsnummer.

I N H O U D

pag 2	Bestuur Atom computerclub	
pag 2	Clubwinkel	
pag 3	Inhoud Atom Nieuws	
pag 3	Atom-markt	
pag 4	Regioschijf 4 - 1990	
pag 4	Regio-mededelingen	
pag 5	Van de redaktie	W.Truijen
pag 6	6502 - Tracer	J.Swinkels
pag 25	Atom aan de Pc	M.van Leuven
pag 29	S P S Printservice	
pag 30	Wat doet de VIA ?	J.Lernout
pag 36	Bericht v.d. Federatie	
pag 37	Drukker	J.Lernout
pag 39	Mededeling	
pag 41	Indeling #B800 - gebied	B.Tossaint
pag 45	Electron als printerbuffer	R.Leurs
pag 50	Modelbaanbesturing met de Atom	R.Leurs
pag 54	POWER-SPIN beschrijving	R.Leurs
pag 59	Regionale besturen	

ATOM NIEUWS is een uitgave van de federatie Atom computerclub Ned/Belgie en verschijnt 4 maal per jaar.
De redaktie gaat er van uit dat de ingezonden copy gemaakt is door de inzender, tenzij in de publicatie uitdrukkelijk anders is vermeld.
De aansprakelijkheid echter betreffende de auteursrechten ligt zonder enig voorbehoud volledig bij de inzender.

A T O M - M A R K T

Nog enkele MDCR's te koop.

2 MDCR's compleet met voeding alles samen in een professionele kast ingebouwd, voorzien van snoer en busconnector voor de Atom.
Prijs : fl. 75.-- exclusief verzendkosten.

Tevens nog enkele losse MDCR's met kap en voorfront.
Prijs : fl. 25.-- exclusief verzendkosten.

Bij iedere bestelling, en dat zolang de voorraad strekt, worden er gratis 5 lege bandjes meegeleverd.

ATOM-NIEUWS REGIO-SOFTWARE '90

dnr	program	geheugen#	utility	soort	artikel	info
05	$XMODEM	2800-3826	--	UTILITY	MODEM	PC AAN DE ATOM
05	$xmodem	2800-3972	--	UTILITY	MODEM	PC AAN DE ATOM
05	2BASC28	2800-2900	--	SPEL	POWER	BASIC VOOR SPEL
05	2BASC65	6500-6800	--	SPEL	POWER	BASIC VOOR POWERSPIN
05	2BASC98	9800-A000	--	SPEL	POWER	BASICPROGRAMMA VOOR SPEL
05	2BONS68	6800-8000	--	SPEL	POWER	SCHERM VOOR SPEL
05	2CONTROL	8000-9800	--	SPEL	POWER	SCHERM VOOR SPEL
05	2MACH58	5800-6500	--	SPEL	POWER	MACHINETAAL VOOR SPEL
05	2POWER	8000-9800	--	SPEL	POWER	SCHERM VOOR SPEL
05	2PUNTEN	8000-9800	--	SPEL	POWER	SCHERM VOOR SPEL
05	2SCHM80	8000-9800	--	SPEL	POWER	SCHERM VOOR GOKKAST
05	DRUKKER	5000-5FB4	--	UTILITY	VIA	PRINTERBESTURING
05	DXMODEM	2800-3A47	--	UTILITY	MODEM	PC AAN DE ATOM
05	GKKST-2	2900-4F00	--	SPEL	POWER	GOKKAST VERSIE 2
05	REVERS+	2900-4200	-	SPEL	--	SPEL UIT DE OUDE DOOS
05	SXMODEM	2800-4071	--	UTILITY	MODEM	PC AAN DE ATOM
05	TTY6551	2800-5C52	--	UTILITY	MODEM	PC AAN DE ATOM
05	UXMODEM	2800-3C72	--	UTILITY	MODEM	PC AAN DE ATOM
05	VDU	5000-5800	--	SPEL	POWER	HULPFILE POWERSPIN
05	XMODEM	6000-61BF	--	UTILITY	MODEM	PC AAN DE ATOM

REGIO-MEDEDELINGEN.

REGIO LIMBURG-BELGIE

Clubavonden in "Oos Kaar",Geldersestraat 43,tel 04490-21378
Data : 5 October, 2 November, 7 December.

Progr. 5 Oct. : DA conversie, d.m.v. klein printje met enkele onderdelen; demo daarvan inclusief uitleg; printje met onderdelen alleen verkrijgbaar op de clubavond.

Progr. 2 Nov. : Stappenmotoren, mogelijkheden, techniek, demo; mogelijkheden in modelbouw. discussie over andere toepassingen.
Eerste discussie over ontwerp "MINI-ATOM".

Van de redaktie

Ter gelegenheid van een van de regio-clubavonden werden er enkele vragen gesteld voor wat betreft "Het Bronsgroen Eikeltje" op disk.
Allereerst iets over de geschiedenis van het "Diskette Eikeltje.
Het initiatief tot het uitbrengen van deze diskette was ontstaan na de jaarlijkse ledenvergadering in december 1989.
Enkele knappe koppen van onze club wisselden van gedachte over de mogelijkheid tot het uitgeven van een diskette in de vorm van een Uptime-disk (C). Deze heren, H.Bastings en R. Leurs, zijn toen gaan overleggen in welke vorm zoiets gegoten moest worden; welke soort software geschikt zou zijn om op deze manier te publiceren.
Tijdens een van de volgende clubavonden werdt al het eerste exemplaar gepresenteerd en ter beoordeling aangeboden.
Hier en daar werden wat suggesties gedaan waar de auteurs weer mee aan de gang konden gaan. Een van de voorwaarden was dat alle ,op de schijf voorkomende software, ook moest kunnen werken op de 80-kolom's kaart.
Het gevolg hiervan was een extra beeldschermroutine met 24 regels en 40 kolommen op het scherm.
In begin van de maand maart was het dan zover; het tweede exemplaar was nu een feit. Het was mogelijk om in 24/40 mode te werken alsook in 80 kolom's mode. De volgende opgave bestond er in om clear-4 schermen in te laden om de presentatie wat te verfraaien. Na wat geexperimenteer leverde dat dus uiteindelijk "Het Bronsgroen Eikeltje" versie 3 op.
Dit was dus in het kort, en zeker niet volledig, de geschiedenis van ons "Uptime Eikeltje". Omdat de twee eerste uitgaven, volgens de auteurs, niet geschikt waren voor verspreiding; zijn ze dus ook niet via de clubsoftware verdeeld ze zijn wel ter beschikking in het DISK-archief. Belangstellenden kunnen zich ten alle tijde melden bij de beheerder W.Truijen, waarvan het adres in elke uitgave van Atom-nieuws te vinden is.
Om nu nog even terug te komen op te toekomst van het "schijf-Eikeltje"; deze is geheel afhankelijk van de enthoesiaste medewerking van onze clubleden. Daarom bij deze alsnog een oproep om uw programma's en artikelen aan de redaktie te bezorgen. Voor eventuele herwerking en aanpassing van de begeleidende teksten kan gezorgd worden.
Het is zeker niet zo, dat er met de Atom niets meer te beleven valt en aan de reacties en commentaren tijdens de clubavonden kunnen we afleiden dat er bij de leden nog vele ideeen sluimeren.

DEEL ZE MEE AAN DE REDACTIE !!!

Met vriendelijke groeten van uw diskarchivaris.

Geldrop, 21-05-1990.

Beste Atomisten,

Allereerst wil ik aan iedereen, die mijn hardware-tracer heeft na gebouwd,mijn verontschuldiging aanbieden. Toen mijn systeem werkte, die ik met behulp van mijn tracer aan de praat heb gekregen, ben ik er nl. van uitgegaan dat de tracer perfect werkte. Niets is minder waar! Ik heb me nl. ONBEWUST NIET aan de data-sheet specificatie van de 6502 processor gehouden!

voetnote van de data-sheet
... wat een kluns, kan niet eens lezen wat duidelijk voor zijn neus geschreven staat, die denkt ook overal verstand van te hebben....

In de schakeling wordt de ready-lijn synchroon met fi2 laag gemaakt maar ASYNCHROON d.m.v. een flip-flop weer hoog gemaakt en dat MAG NIET!!!

De ready- lijn mag ALLEEN laag of hoog worden gemaakt indien fi1 een OPGAANDE flank heeft of reeds hoog IS!!!! Door schade en schande ben ik hier zelf al achter gekomen. Echte hardware Atomers hebben mij hier ook al op attent gemaakt. Voor hen geldt echter dat ook deze oplossing te vergeefs is.

IK HEB NL. EEN GEHEEL NIEUWE STEPPER ONTWIKKELD DIE NU PERFECT WERKT IN ZOWEL SYNC-MODE (INSTRUCTIE-CYCLUS VOOR INSTRUCTIE-CYCLUS) ALS IN FI-MODE (CLOCKPULSE VOOR CLOCKPULSE)!!!!!

Hierbij is het NU mogelijk om ZOWEL de OPCODE als ook de OPERAND te bekijken.!!! Dit houdt dus in dat je niet alleen ziet naar welk adres wordt gesprongen maar je ziet nu ook WAT daar gebeurt.

De nieuwe stepper werkt nl. op het principe van CLOCK-STRETCHING.

Bij deze wil ik dan ook John Feron en Henk Bastings bedanken voor hun broodnodige support die ik van hun heb mogen ontvangen. Zonder hun kritische opmerkingen was ik nooit uit de vizieuze cirkel gekomen waar ik weken in heb rond gedraaid.

Op de volgende pagina's volgt de beschrijving van de nieuwe hardware-tracer inclusief schema's voor de fanatieke bouwer's onder ons.

1.1 Waarom een nieuwe stepper-schakeling ?

Waarom ik een nieuwe tracer heb ontwikkeld is zeer simpel.
De vorige tracer was niet in staat om op ELK gewenst moment te starten!!!! Door synchroon de ready-lijn hoog te maken ben je op GEEN enkele manier in staat om te achterhalen of DAADWERKELIJK de processor gestart is bij die ENE startpulse, die gegeven is.

Verder was ik NIET in staat om de processor precies te volgen (clock-pulse voor clock-pulse).
Deze, voor mij zeer belangrijke trace-mogelijkheid, was voor mij de doorslag om het roer niet 90 graden maar 180 graden om te draaien en een GEHEEL NIEUWE stepper te ontwikkelen. Door de vele telefoontjes en moeilijke gesprekken met vrienden en bekenden viel plotseling de kreet CLOCK-STRETCHING. Hiermee wordt bedoeld dat je de clock die de processor bestuurd een klein beetje vertraagt waardoor de langzame randapparatuur de kans krijgt om te doen wat van hem verlangt wordt. Dat klein beetje vertragen heb ik natuurlijk weer tot in het uiterste doorgetrokken en heb nu een clockbesturing gemaakt waarbij ik de clock niet een klein beetje vertraag wordt maar 27 uur!!!!

Zoals in de data-sheets van de 6502 processor te lezen valt is het niet mogelijk om de clock-pulse op een lagere frequentie te zetten dan 850 KHz omdat dan de interne DYNAMISCHE registers van de processor hun waardes verliezen. (Heb ik uitgeprobeerd en klopt.) Dit komt omdat dan de clock veel te lang LAAG blijft waardoor de data verloren gaat. Echter wat gebeurt er als de clock veel te lang HOOG wordt gehouden ? Volgens de geleerde boeken, en ik heb er heel veel, mag clock-stretching niet langer dan 2,1 uSec. duren i.v.m. refresch van data in dynamisch registers. Ik ben zo eigenwijs om uit te proberen wat er dan WEL gebeurt als je de clock hoog HOUDT...!!!!! En zie hier de truck met de processor.... de processor staat STIL... de data gaat NIET verloren in de registers want na 27 uur start ik de processor weer op (welke idioot probeert dit ook!) en jawel grote Atomisten .. de processor gaat gewoon VERDER met het werk waar hij 27 uur eerder mee gestopt was...... (hiep hiep hoeraaaaaaa....)

Nadat ik deze truck had ontdekt was het toen de beurt aan mij om een schakeling te maken die de clock fi0 ging besturen. Na veel koffie en slapeloze nachten heb ik dit karwei weer voor elkaar gebokst en hieronder volgt het resultaat.

* ADRESS DETECTED

Vcc

3K3

PA D5

stop cpu

74HCT132

74LS74 1B

Q_nc

alfa

enable / disable single step

alfa not

74123 B

jumper

* start cpu PA D4

stop at sync

74HCT132

74LS74 2A

alfa 2

PI-8 ??? CPU

start na sync stopped cpu

disable sync

3K3

74HCT132

74LS74 1A

alfa not

SYNC

74123 A

74HCT132

74LS74 2B

nc

alfa not

alfa

74LS132

74LS132

74LS132

74LS132

alfa 2

10nF

t=0 t=8 nS.

t=47 nS. (tyden gemeten!)

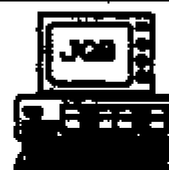
C dient ervoor om een goede NUL van je clock te krijgen!

JGB SYSTEMS

Jan Carstensweg 3?3

5665 TJ Geldrop

Tel : 040 - 85 23 25



Title

1SSTEP.SCH

Size	Document Number	REV
A	1	0

Date: January 1, 1990 Sheet of

TIJD DIAGRAM CLOCK-PULSEN

fi-0 from 74LS383 = alfa

alfa not

alfa 1

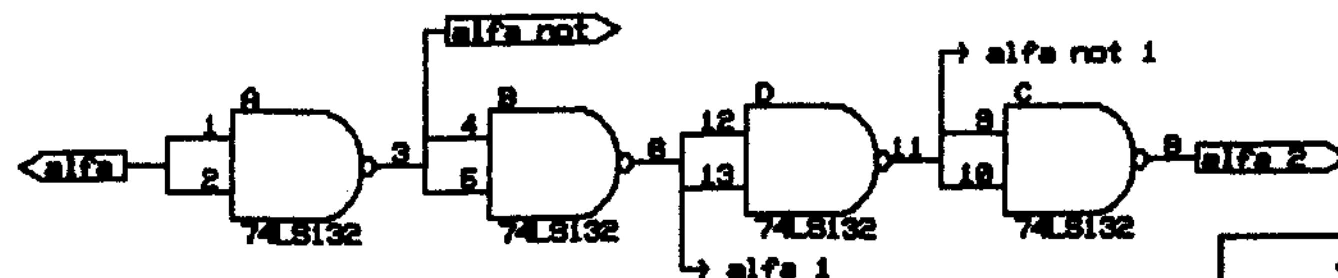
alfa not1

alfa 2

FI-0 to processor

0 9 1 2 3 4
8 8 8 7

tijden gemeten in nS.



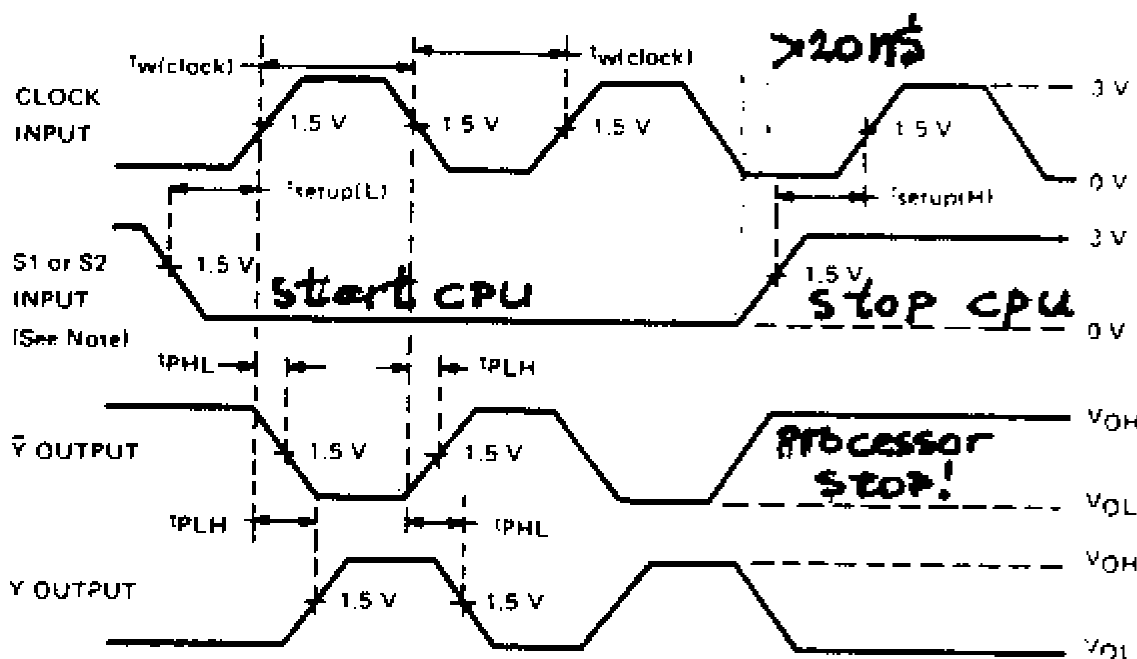
JGS SYSTEMS Jan Carstensweg 288 5665 TJ Geldrop Tel : 040 - 85 23 28		
Title CLOCK.TYD		
Size A	Document Number 1	REV 0
Date: May 24, 1988	Sheet of	

2.0 Uitleg werking IC 74120 dual clock-synchronizer/driver.

Als intro zal ik maar eerst een kleine uitleg geven hoe het clock-ic 74120 precies werkt. In figuren 2,4 en 5 zijn de tijd-diagrammen te zien van de besturings-signalen M, S1 en S2 van het ic. Het principe is erg eenvoudig (hoe is het mogelijk na zo'n story)....

2.1 Processor starten.

Op de clock-ingang (clk-in) wordt het 1 of 2 Mhz signaal aangeboden dat door de 4-deler (74ls393) aangemaakt wordt. De besturings-signalen M,S1 en S2 zorgen er nu voor dat GEHEEL SYNCHROON met de clock-ingang (clk in) het UITGAANDE clock-signaal fi0 (clk out) voor de processor wordt gegenereerd. Indien M en S1 laag zijn en S2 hoog is, zal de uitgaande clock-pulse (clk out) de geinverteerde clock-pulse zijn, welke aan de ingaande clockingang (clk-in) aangeboden wordt.
 ---> processor loopt op 1 of 2 MHz.....



Mode control and R Inputs are low unused S input is high.

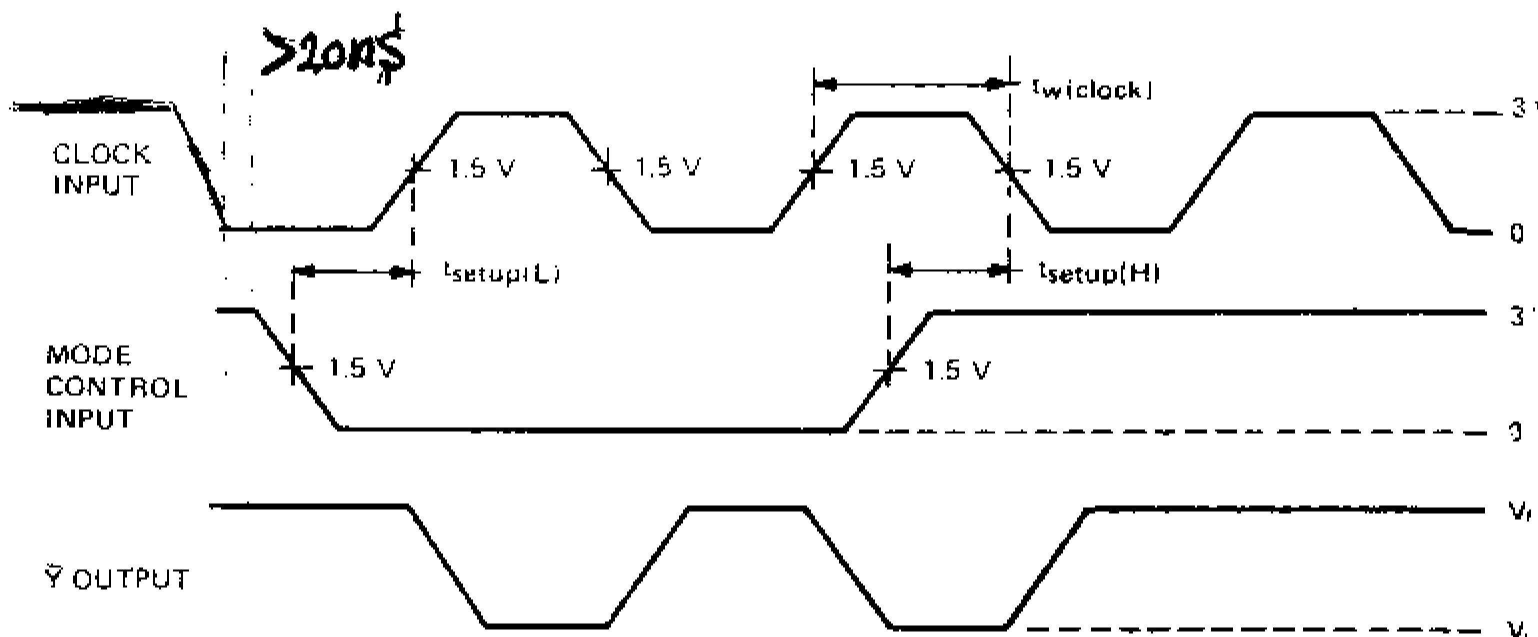
FIGURE 2–INITIATING AND TERMINATING PULSE TRAIN FROM S INPUTS

2.2 Processor stoppen.

Wordt nu signaal S1 hoog gemaakt, dan zal op de neergaande flank van clk-in INTERN de uitgaande clock-pulse HOOG worden en hoog BLIJVEN ----> de processor stopt........ We maken dus NU gebruik van CLOCK-STRETCHING. Op DIT moment is er dus ALLE tijd op de gehele schakeling door te meten bv. systeem-bussen en adres-decodering ....etc.

TYPES SN54120, SN74120
DUAL PULSE SYNCHRONIZERS/DRIVERS

PARAMETER MEASUREMENT INFORMATION



NOTE: At least one of the S inputs is low.

FIGURE 4—INITIATING AND TERMINATING PULSE TRAIN WITH MODE CONTROL INPUT

2.3 Processor starten NA stop

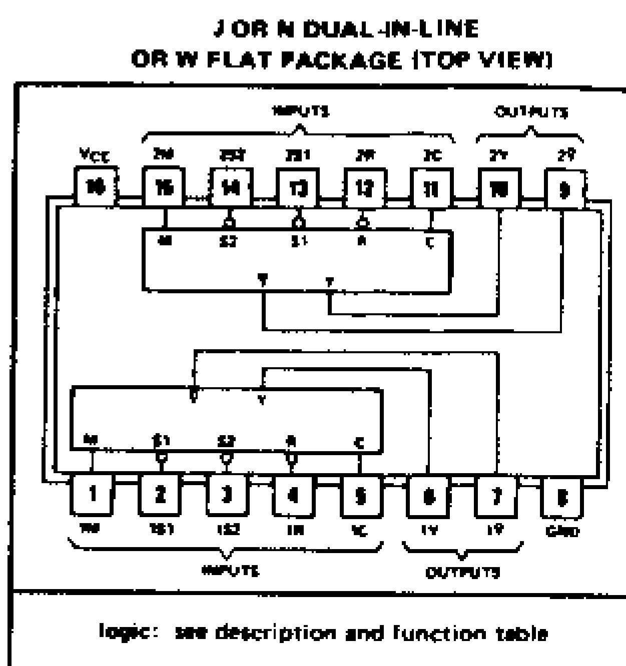
Wordt bv na 27 uur S1 weer laag gemaakt, dan zal intern het hele zaakje synchroon worden doorgeschakeld. Doordat S1 laag gemaakt is, zal bij de eerst volgende overgang van clk-in de clk-in worden doorgeschakeld naar clk-out waardoor de uitgaande clock de processor weer zal starten ----> de processor werkt weer op volle toeren. We kunnen dus de processor stoppen en starten door slechts EEN signaal naar het ic hoog OF laag te maken. Hoe we dit signaal aanbieden is helemaal niet van belang. Je kunt dit met een dure PC doen of met een heel simpel goedkoop schakelaartje!! Intern wordt alles geregeld.

note !!!!

> Het enige wat NIET mag is het aanbieden van de besturings-signalen 20 nano-seconden NADAT clk-in van hoog naar laag gaat. Wordt dit wel gedaan dan is het niet gegarandeerd dat de clk-out METEEN wordt bestuurd of EEN clock-pulse later!

- Generates Either a Single Pulse or Train of Pulses Synchronized with Control Functions
- Ideal for Implementing Sync-Control Circuits Similar to those Used in Oscilloscopes
- Latched Operation Ensures that Output Pulses Are Not Clipped
- High-Fan-Out Complementary Outputs Drive System Clock Lines Directly
- Internal Input Pull-Up Resistors Eliminate Need for External Components
- Diode-Clamped Inputs Simplify System Design
- Typical Propagation Delays:

  9 Nanoseconds through One Level
  16 Nanoseconds through Two Levels



FUNCTION TABLE

INPUTS			FUNCTION
R	S1	S2	
X	L	X	Pass Output Pulses
X	X	L	Pass Output Pulses
L	H	H	Inhibit Output Pulses
H	↓	H	Start Output Pulses
H	H	↓	Start Output Pulses
↓	H	H	Stop Output Pulses
H	H	H	Continue†

H = high level (steady state)

L = low level (steady state)

↓ = transition from H to L

X = irrelevant

†Operation initiated by last ↓ transition continues

2.4 Op juiste moment aanbieden van S1,S2 en M signalen.

Om aan deze specificatie te voldoen zorg ik ervoor dat het clock-in signaal wordt opgedeeld in meerdere clock-pulsen die ieder op zich een delay t.o.v. elkaar hebben. Hierdoor ben ik dan in staat om het ic op de juiste manier te besturen. De besturings-signalen S1,S2 en M worden eerst naar een flip-flop gestuurd die de signalen pas doorlaat naar het IC 74120, indien clk-in hiervoor de opdracht geeft --> op juiste tijdstip doorklokken. Hierdoor wordt dus VOORKOMEN dat asynchrone signalen op het VERKEERDE tijdstip de 74120 besturen. Door deze constructie te kiezen is de besturing van het ic geheel onafhankelijk geworden van de manier WAAROP het ic bestuurd wordt. Op ieder tijdstip kunnen nu de signalen worden aangeboden. clk-in zorgt er ZELF voor dat de signalen NIET VERKEERD worden doorgegeven. Dit lijkt om- slachtig en moeilijk maar is wel precies volgens de data-sheet-specificatie's en dus FOUTLOOS!

TYPES SN54120, SN74120
DUAL PULSE SYNCHRONIZERS/DRIVERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)		7 V
Input voltage		5.5 V
Interemitter voltage (see Note 2)		5.5 V
Operating free-air temperature range:	SN54120 Circuits	−55°C to 125°C
	SN74120 Circuits	0°C to 70°C
Storage temperature range		−65°C to 150°C

NOTES 1. Voltage values, except interemitter voltage, are with respect to network ground terminal.
2. This is the voltage between two emitters of a multiple-emitter transistor. For this circuit, this rating applies between the S1 and S2 inputs.

recommended operating conditions

			SN54120 MIN	SN54120 NOM	SN54120 MAX	SN74120 MIN	SN74120 NOM	SN74120 MAX	UNIT
Supply voltage, V_{CC}			4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}					−2.4			−2.4	mA
Low-level output current, I_{OL}					48			48	mA
Setup time (see Figures 2 thru 5)	Any input except mode control, $t_{setup(H\ or\ L)}$		12			12			ns
	Mode control	$t_{setup(H)}$	0			0			
		$t_{setup(L)}$	12			12			
Hold time (see Figures 3 and 5)	Any input except mode control, $t_{hold(H\ or\ L)}$		3			3			ns
	Mode control, $t_{hold(H\ or\ L)}$		20			20			
Operating free-air temperature, T_A			−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage		V_{CC} = MIN, $I_I = -12$ mA			−1.5	V
V_{OH}	High-level output voltage		V_{CC} = MIN, $V_{IH} = 2$ V, $V_{IL} = 0.8$ V, $I_{OH} = -2.4$ mA	2.4	3.4		V
V_{OL}	Low-level output voltage		V_{CC} = MIN, $V_{IH} = 2$ V, $V_{IL} = 0.8$ V, $I_{OL} = 48$ mA		0.2	0.4	V
I_I	Input current at maximum input voltage		V_{CC} = MAX, $V_I = 5.5$ V			1	mA
I_{IH}	High-level input current	Clock input	V_{CC} = MAX, $V_I = 2.4$ V			80	µA
		Other inputs		−0.12	−0.2	−0.36	mA
I_{IL}	Low-level input current	Clock input	V_{CC} = MAX, $V_I = 0.4$ V			−3.2	mA
		Other inputs				−2.1	
I_{OS}	Short-circuit output current§		V_{CC} = MAX	−35		−90	mA
I_{CC}	Supply current		V_{CC} = MAX, See Note 3		55	80	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at $V_{CC} = 5$ V, $T_A = 25°C$.
§ Not more than one output should be shorted at a time.
NOTE 3. I_{CC} is measured with ground applied to all inputs except R which is at 4.5 V and all outputs open.

switching characteristics, $V_{CC} = 5$ V, $T_A = 25°C$

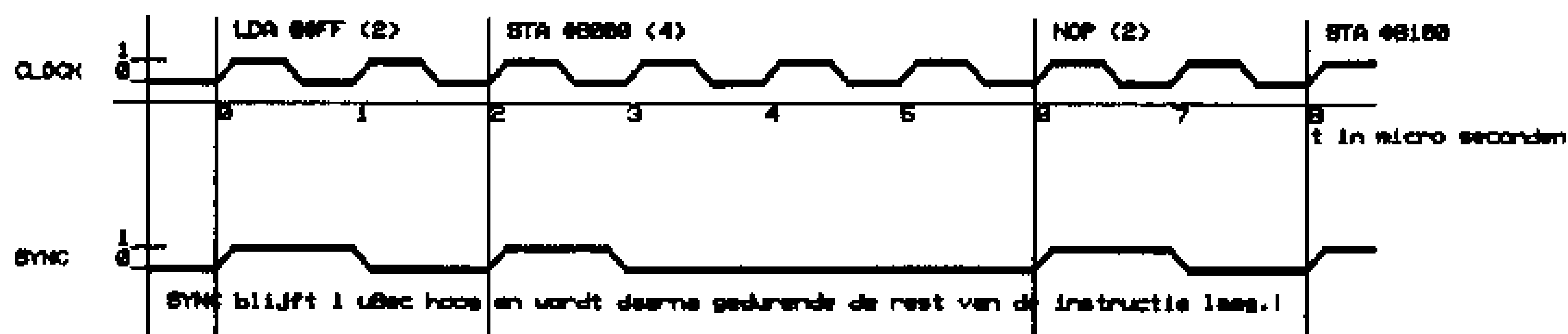
PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	C	Y	$C_L = 45$ pF, $R_L = 133\ \Omega$, See Figure 1		14	22	ns
t_{PHL}					17	25	
t_{PLH}	C	$\bar{Y}$			10	18	ns
t_{PHL}					8	13	

¶ t_{PLH} ≡ Propagation delay time, low-to-high-level output
t_{PHL} ≡ Propagation delay time, high-to-low-level output

2.5 Processor in Sync-mode.

In voorgaande hoofdstukken heb ik proberen uit te leggen dat ik in staat ben geweest om de processor te starten, te stoppen en weer op te starten. Om nu de stepper helemaal te optimaal te kunnen benutten ga ik de processor nu stoppen als deze een opcode-fetch gaat plegen (begin van een NIEUWE instructie!).
Zoals reeds bekend is wordt een signaal naar de 74120 gestuurd waardoor de processor stopt/start. Door nu het SYNC-signaal van de processor te gebruiken als trigger, wordt dus de processor gestopt bij het BEGIN van zijn instructie! De processor zet zich dus ZELF STIL! Omdat de processor stil staat weet ik nu ook welk niveau het sync-signaal heeft. Dit was bij de vorige steppen NIET het geval omdat je niet precies wist op WELK moment de processor eigelijk stopt. Het sync-signaal wordt hoog op het moment dat de processor een fetch pleegt en dit signaal blijft gedurende EEN volledige clock-pulse van fi0 hoog! Na deze clock-pulse wordt het sync-signaal laag totdat er weer en opcode-fetch wordt uitgevoerd. Zie onderstaand figuur over werking sync-signaal.

TYD-DIAGRAM CLOCK SIGNAAL EN SYNC SIGNAAL.



Als de processor stil staat, en dus het sync-signaal stabiel is, kunnen we door het signaal te disabelen de processor weer door laten lopen. Bij de eerst volgende opcode-fetch zal het sync-signaal weer hoog worden en de processor automatisch op het juiste moment stoppen. We kunnen dus de processor in SYNC-MODE door een programma laten steppen. Deze sync-mode houdt dus in dat je ALLEEN het adres en opcode te zien krijgt, hetgeen in sommige gevallen lastig kan zijn.

* ADRESS DETECTED

Vcc
3K9
Pb D5
stop cpu
A
74HCT132
Vcc
D PR Q
CLK
CL Q̄
6 nc
alfa
74LS74 1B
Vcc
enable / disable single step
alfa not
Vcc
M C B2 B1 R Y Y
10 nc
74120 B
Vcc
3K9
jumper
C
* [redacted]
start cpu Pb D4
stop at sync
74HCT132
Vcc
alfa 2
74LS74 2A
8 nc
FI=0 6502 CPU

start na sync stopped cpu
disable sync
Vcc
3k9
D
74HCT132
alfa not
Vcc
74LS74 1A
5 nc
Vcc
alfa
Vcc
Vcc
M C S2 S1 R Y Y
74120 A
6 nc
SYNC
74HCT132
Vcc
74LS74 2B
8 nc
6 nc
Vcc

alfa not
alfa
A B D C
74LS132 74LS132 74LS132 74LS132
alfa 2
10nF
t=0
t=8 nS.
t=47 nS. (tyden gemeten!)

C dient ervoor om een goede NUL van je clock te krijgen!

JGS SYSTEMS
Jan Carstensweg 259
5665 TJ Geldrop
Tel : 040 - 86 23 26

Title		
	120STEP.SCH	
Size	Document Number	REV
A	1	0
Date: January 1, 1980	Sheet	of

TIJD DIAGRAM SYNC-MODE 74120 SYNCHRONISER

disable sync

asynchroon gegeven

S1 74120a

ingeklokt door alfa not (synchroon)

alfa

Y(not)74120a

een puls t.v.g.disable sync

SYNC processor

S1 74120b

ingeklokt door alfa2

start cpu t.g.v disable sync

alfa 2

alfa not

FI-0 processor

clock blijft hoog t.g.v. S1 74120b

clock start door SYNC

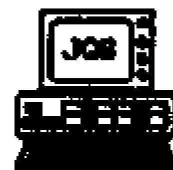
opstarten —> clock loopt

cpu clock synchroon opgestart t.g.v. S1 74120b

NOTE : S2 (3) = HIGH
R (4) = LOW

JGS SYSTEMS

Jan Carstenszweg [illegible]
[illegible] TJ Geldrop
Tel : 040 - [illegible]



Title

SYNC-MODE.TYD

Size	Document Number	REV
A	1	0

Date: May 24, [illegible] Sheet of

2.6 Processor in STEP-MODE.

Het is soms gemakkelijk dat je PRECIES weet wat de processor uitvoert. Denk maar eens aan het ophalen van vector-adressen uit zero-page als je niet weet of deze wel goed staan. Door nu in de step-mode door een programma te fietsen ben je ook in staat om te zien welke data er op een bepaald adres wordt weggeschreven op opgehaald.

Als we de processor in step-mode zetten maken we gebruik van een combinatie van twee besturings-signalen die we naar de 74120 zenden.
Allereerst zetten we de processor gewoon stop door S1 hoog te maken. DAARNA!!! maken we signaal M hoog, waardoor intern de clock gelocked wordt. Als we S1 weer laag maken NADAT signaal M hoog is gemaakt, dan zal het IC ervoor zorgen dat GEHEEL SYNCHROON met clk-in op de uitgang clk-out EEN VOLLEGIGE clock-pulse verschijnt, waarna clock-out weer hoog BLIJFT! Dit heeft dan als gevolg dat de processor slechts EEN clock-pulse voor zijn kiezen krijgt en dus het programma maar een clock-pulse verder afhandeld.
Staat er een instructie bv. STA #8000 (4 clock- pulsen), dan zullen we deze combinatie's van signalen dus ook vier maal moeten uitvoeren.

Bij - eerste pulse wordt opcode #8D opgehaald,
      tweede pulse wordt operand #00 opgehaald,
      derde pulse wordt operand #80 opgehaald,
      vierde pulse wordt inhoud accu op #8000 gezet

Je ziet dus bij de vierde pulse daadwerkelijk dat er op adres #8000 wordt geschreven maar ook WAT er wordt geschreven. Natuurlijk kun je nu ook wel indenken dat je bij een lees-instructie ook kunt zien WAAR de processor inleest en WAT precies!

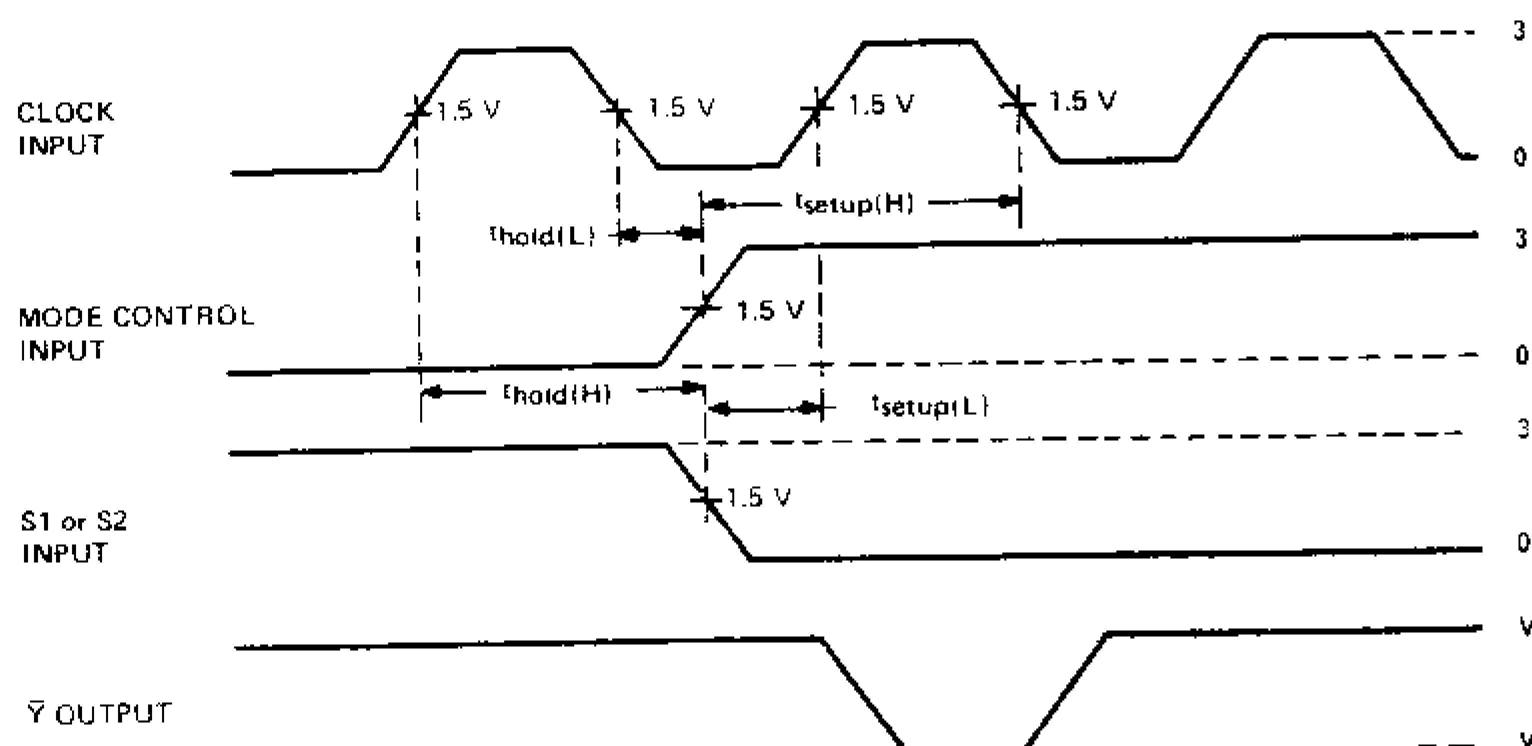


FIGURE 5—ENABLING SINGLE PULSE

* ADRESS DETECTED

Vcc
3K9
PB D6
stop cpu
A
1 2
1
3
74HCT132
Vcc
12
D PR Q
9
11
CLK
alfa
CL Q̄
8 nc
74LS74 1B
Vcc

enable / disable single step

15
M C
alfa not
Vcc
S2 S1 R
Y̅
9
Y
10 nc
74120 B

Vcc
3K9
jumper
* [redacted]
start cpu PB D4
stop at sync
C
9 10
1
8
74HCT132

start na sync stopped cpu
disable sync

Vcc
4
2
D PR Q
5
3
CLK
alfa 2
CL Q̄
6 nc
74LS74 2A
1
IRQ 6502 (CPU)

SYNC
4
B
1
6
5
74HCT132
7

alfa
Vcc
M C
Y̅
Vcc
S2 S1 R
Y
8 nc
74120 A

Vcc
3K9
D
12
13
1
11
74HCT132
alfa not
Vcc
4
2
D PR Q
5 nc
3
CLK
CL Q̄
6
74LS74 1A
1
Vcc

Vcc
12
D PR Q
9 nc
11
CLK
CL Q̄
8 nc
74LS74 2B
13
Vcc

alfa not
alfa
A
1 2
2
3
74LS132
B
4 5
2
6
74LS132
D
12 13
2
11
74LS132
C
9 10
2
8
alfa 2
74LS132
10nF
t=0 t=9 nS.
t=47 nS. (tyden gemeten!)

C dient ervoor om een goede NUL van je clock te krijgen!

JGS SYSTEMS
Jan Carstensweg 258
5665 TJ Geldrop
Tel : 040 - 86 23 28

JGS

Title		
128STEP.SCH		
Size	Document Number	REV
A	1	0
Date: January 1, 1980	Sheet	of

TIJD DIAGRAM STEP-MODE 74120 SYNCHRONISER

stop cpu asynchroon gegeven

alfa

M 74120b — ingeklokt door alfa (synchroon)

start cpu

alfa 2

S1 74120b — ingeklokt door alfa2

alfa not

FI-0 processor — clock blijft hoog t.g.v. M 74120b

opstarten --> clock loopt

clock one pulse!
H = hoog
S1 = wordt laag!

NOTE : S2 (3) = HIGH
R (4) = LOW

De start en stop signalen duren in werkelijkheid minstens 8 clock-pulsen dus kritisch zijn deze signalen niet!

hier zijn ze VEEL korter getekend i.v.m. plaatsgebrek. Werking is wel te begrijpen.

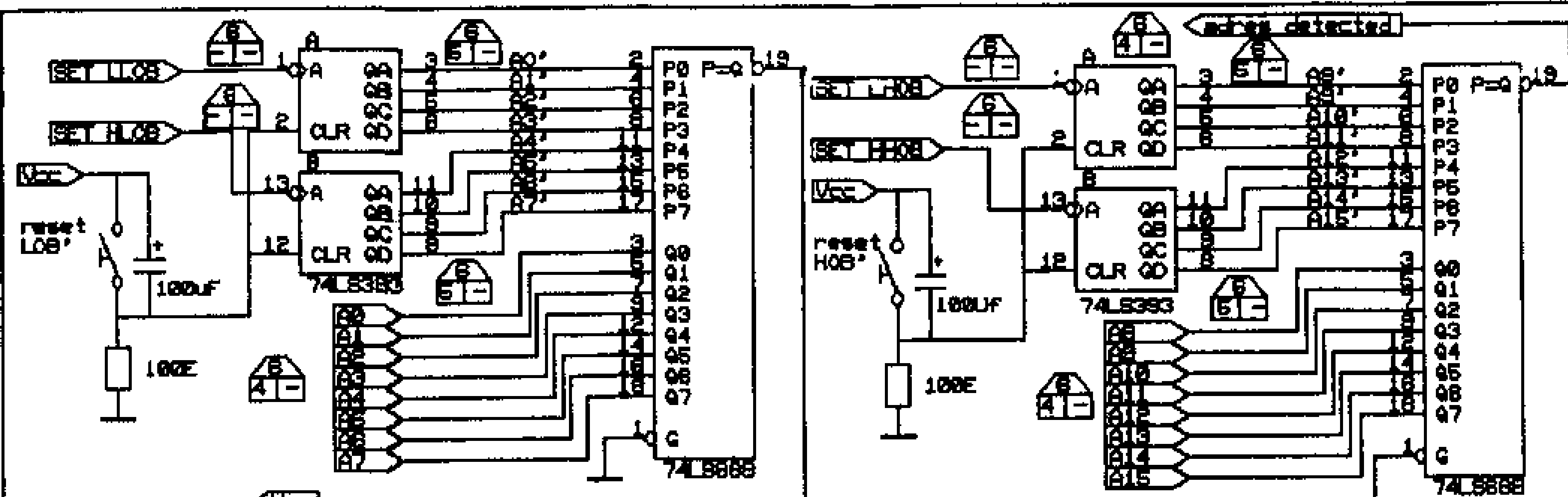
JGS SYSTEMS
Jan Carstenszweg 258
5665 TJ Geldrop
Tel : 040 - 86 23 28

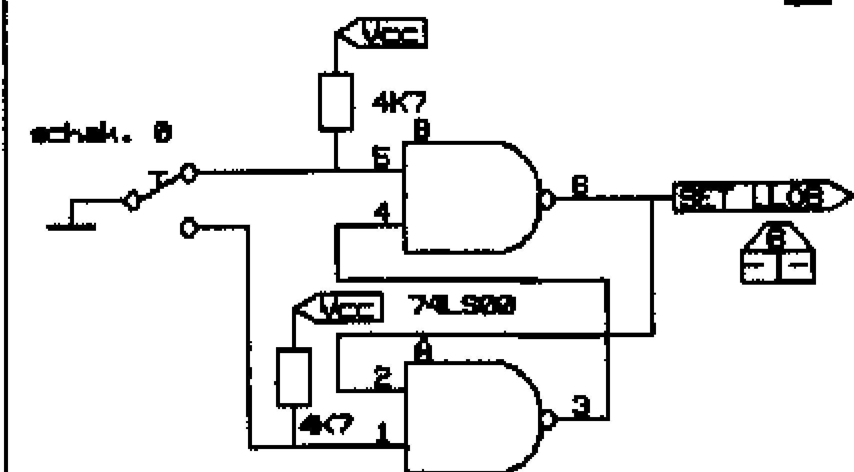


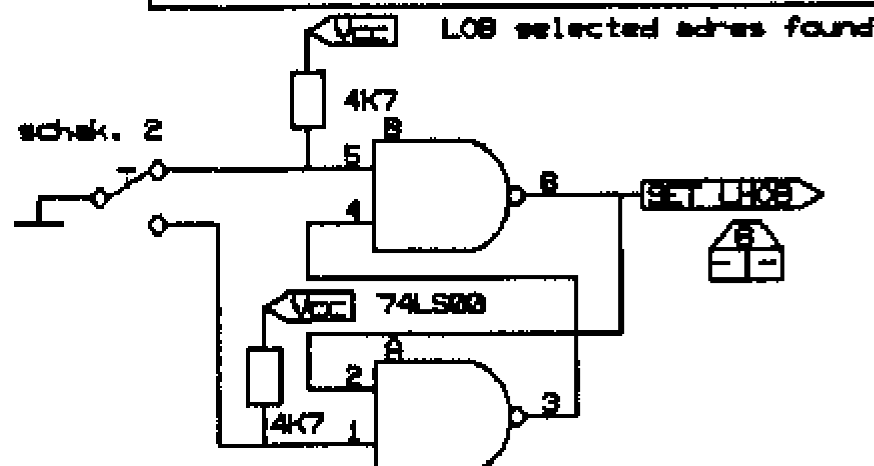
Title		
STEP-MODE.TYD		
Size	Document Number	REV
A	1	0
Date: May 25, 1990	Sheet	of

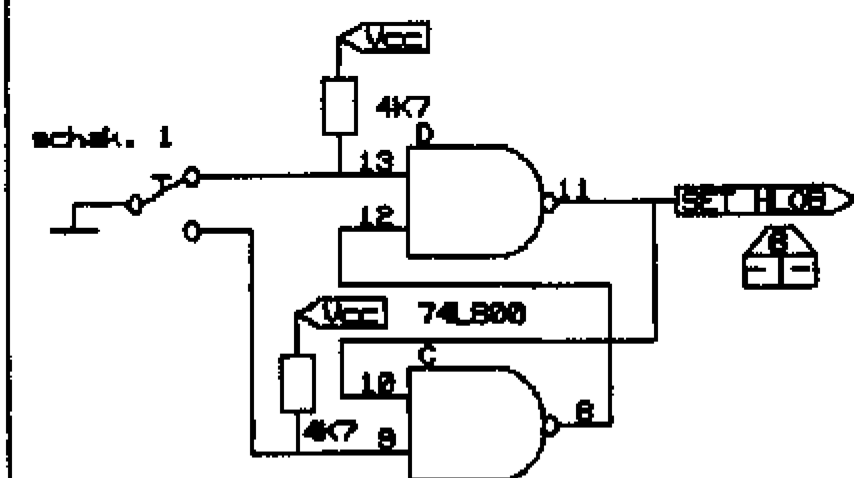
2.7 Find-adres.

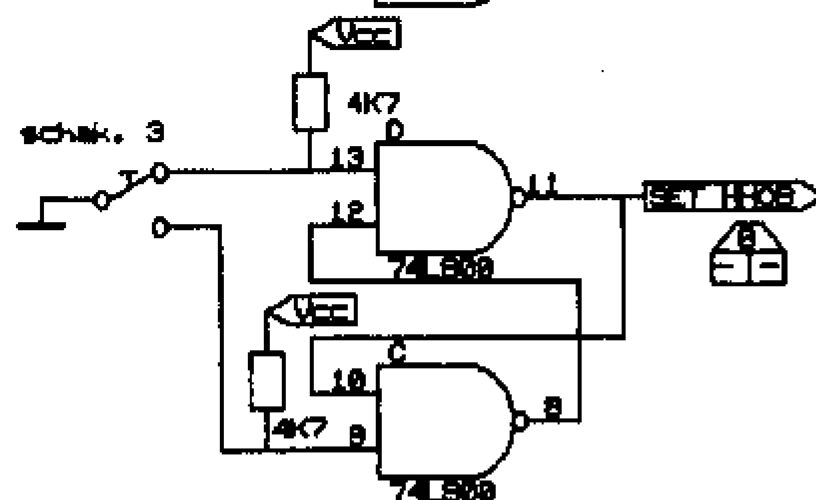
De nieuwe stepper-schakeling is nu helemaal beschreven en uitgelegd. Wat we nu nog niet kunnen doen is het laten stoppen van de processor op ELK WILLEKEURIG ADRES!
Dit is van levensbelang voor een service vriendelijke stepper. Ik moet er niet aan denken dat je steeds door het hele operating-system moet steppen als je alleen je programma op adres #3000 wil bekijken. Om aan deze mogelijkheid het hoofd te bieden heb ik een schakeling ontworpen die continue op de adresbus kijkt of het adres, wat van tevoren ingesteld is, op de bus verschijnt. Door 2 acht-bit comperatoren (74LS688) in cascade te scha- kelen ben je nu in staat om een 16 bits adres te detec- teren. Als het adres wat op de bus verschijnt hetzelfde is als het adres wat staat ingesteld, dan zullen de beide comperatoren een signaal afgeven.Dit signaal zenden we nu naar de 74120 waardoor de processor stopt! De processor stopt dus op het moment dat hij bij het adres is aangekomen, wat we ingesteld hebben. Door DAN de stepper in de STEP- of SYNC-mode te zetten, zijn we in staat om alleen DAAR te steppen waar we behoefte aan hebben. Deze schakeling heb ik nu al ruim een jaar werkend en nog NOOIT ging het fout. De comperatoren kunnen zo snel het adres uitdecoderen waardoor de processor toch nog op het juiste moment stil gezet kan worden. Het instellen van het gewenste adres kun je met jumpers of schakelaars doen. Ik heb echter gekozen voor binaire tellers die het gewenste adres aanbieden aan de comperatoren. Hierdoor ben je veel snelle in staat om het adres te veranderen. Tevens is het goedkoper en heb je minder plaats nodig op je print dan wanneer je schakelaars gebruikt. Wil je verder ook nog hebben dat de data-bus wordt bekeken is het gewoon een kwestie van een comperator erbij. De zin of onzin van deze schakeling zie ik op dit moment nog niet zitten. De schakeling FIND-ADRES heb ik mooi met Orcad getekend en erbij gedaan voor de fanatieke bouwers onder ons.





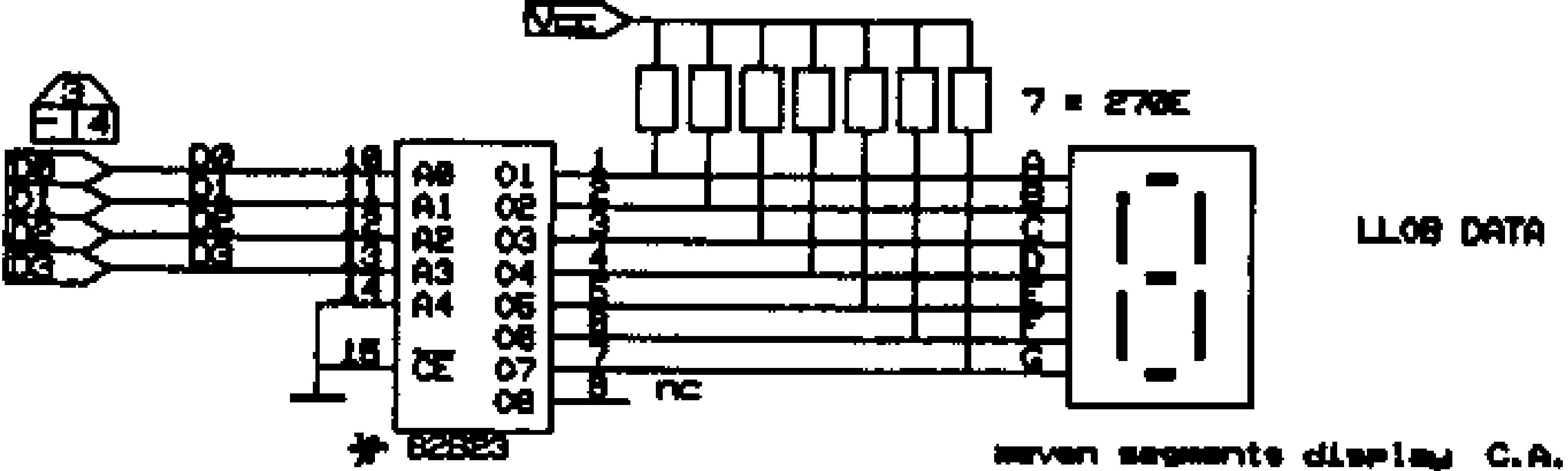


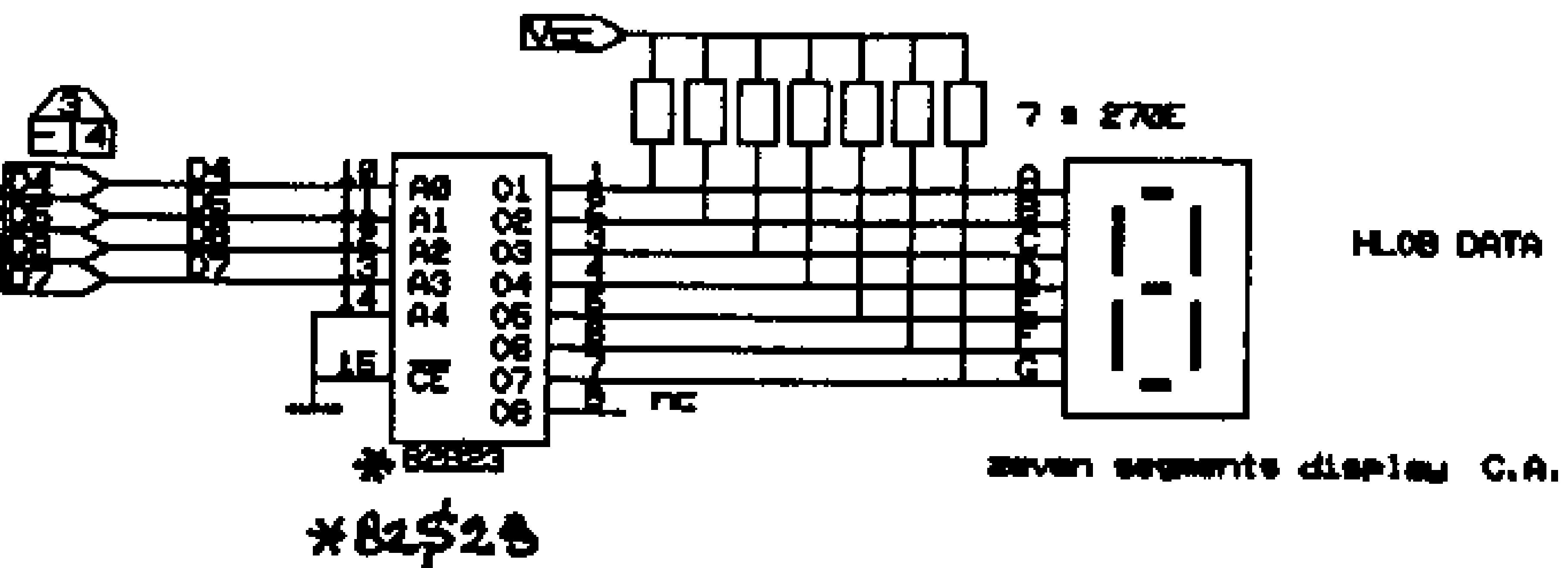




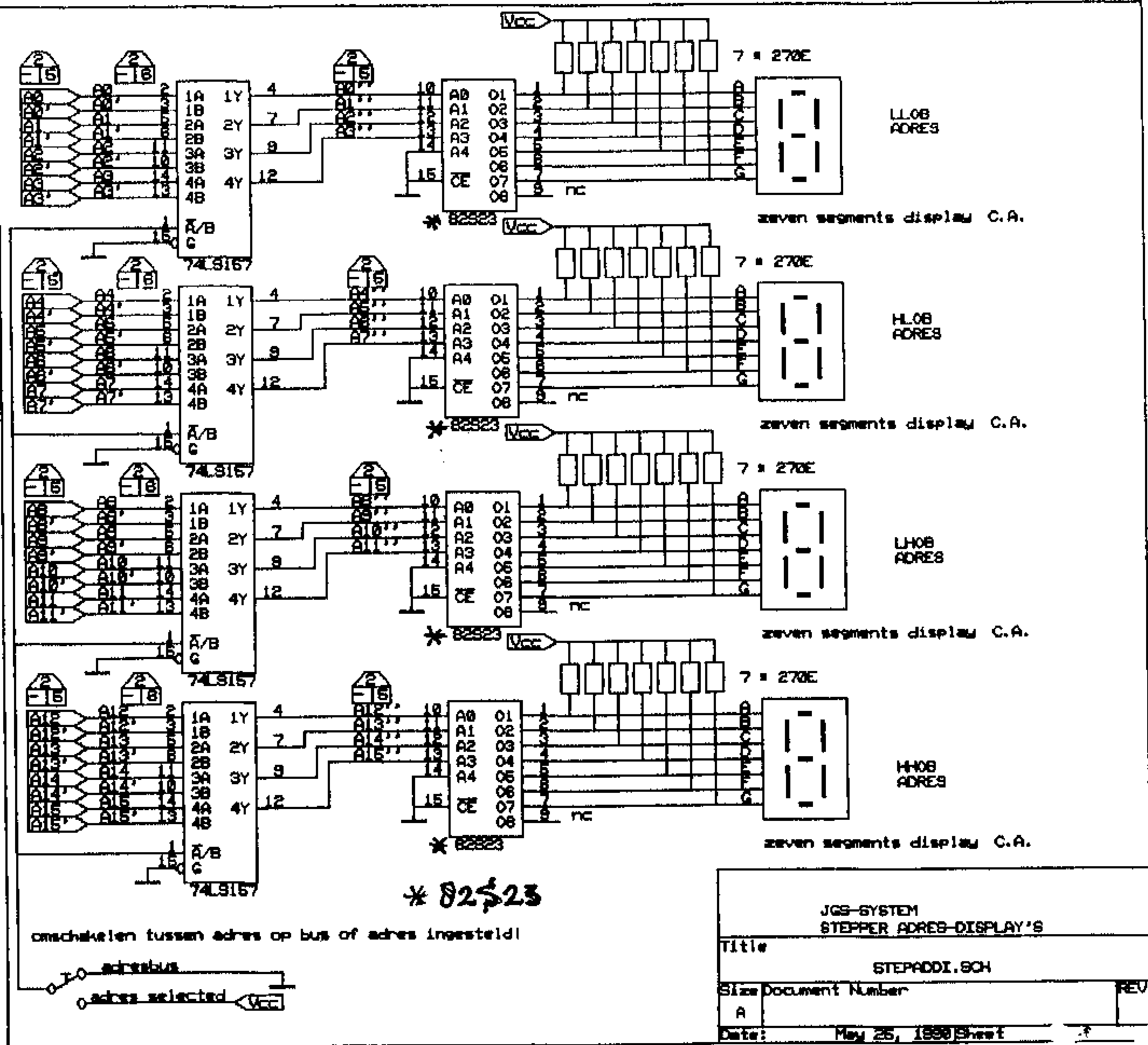
STEPPER ZOEK-ADRES SCHAKELING

Title		
	STEPFIND.SCH	
Size A	Document Number	REV
Date: January 1, 1980	Sheet	of





JGB-SYSTEM STEPPER DATA-DISPLAY'S	
Title STEPDADI.SCH	
Size: A	Document Number	REV
Date: May 25, 1990	Sheet of


7 * 270E
LLOB ADRES
HLOB ADRES
LHOB ADRES
HHOB ADRES
zeven segments display C.A.
74LS157
82S23
Vcc
nc
1A 1B 2A 2B 3A 3B 4A 4B
1Y 2Y 3Y 4Y
A/B
G
A0 A1 A2 A3 A4
CE
O1 O2 O3 O4 O5 O6 O7 O8
* 82S23
omschakelen tussen adres op bus of adres ingesteld!
adresbus
adres selected
JGS-SYSTEM
STEPPER ADRES-DISPLAY'S
Title
STEPADDI.SCH
Size
A
Document Number
REV
Date:
May 25, 1990
Sheet

3.0 Tot slot

Op de Landelijke Atom Dag in de Bilt heb ik dit hele verhaal gedemonstreerd aan alle belangstellenden die daar aanwezig waren. Kritische opmerkingen werden er toen gemaakt over het waarom bouwen van deze "analyser". Er bestaan nl logic-analysers die veel sneller en intellegenter werken dan mijn stepper. Tevens ben je met een analyser real-time metingen aan het verrichten aan je systeem. Looptijdverschillen in de schakelingen haal je er dan snel uit. Met mijn stepper ben je NIET real-time bezig en haal je dus NIET alle fouten eruit! Dit klopt volkomen maar dat was ook HELEMAAL NIET MIJN BEDOELING!!!!
Deze stepper heb ik ontworpen voor metingen te verrichten aan een statisch processor-systeem waarbij adres- en data-signalen stabiel op de bus staan. Hierdoor ben je in staat om de GEHELE HARDWARE NA TE METEN!!!
Dit kun je NIET met een dure logic-analyser.Deze analyser gaat er nl. van uit dat je systeem volledig goed werkt. Zijn er bv. hardware fouten in je systeem zoals verkeerd werkende adres-decoders of verkeerd geprogrammeerde O.S rom's dan zal zo'n dure analyser allemaal flauwe kul gaan inlezen waar je niets aan hebt.
Mijn stepper is dan ook GEEN vervanger van een logic-analyser maar een leuk stukje speelgoed wat gewoon iets anders kan!
Als je met mijn stepper je systeem aan de praat hebt gekregen is het dan nog tijd genoeg om dit duur apparaat uit de kast te halen en dan real-time metingen te verrichten.

Allen die deze stepper na willen bouwen wens ik heel veel succes toe.
Opmerkingen en/of verbeteringen over mijn nieuwste stepper zijn van harte welkom.

Groetjes Jan Swinkels

J. Carstenszweg 259
5665 TJ Geldrop
040 - 86 23 26

J.Feron, Sittard, is genegen en in staat de vereiste PROMS (82S23) te leveren en te programmeren.
tel. 04490- 18847.

Terminal-Emulator

Enkele tijd geleden had ik de behoefte, files die zich op schijven van mijn Atom bevonden naar mijn Pc over te brengen en terug.

Dit moest met zo weinig mogelijk inspanning gebeuren, dwz. met zo weinig mogelijk programmatuur zowel aan de kant van de Atom als van de Pc.

Voor de Pc was er reeds een programma beschikbaar dat toeliet files te transporteren via telefoon naar een bulletin-bord en terug. Dit programma bevat ook een terminal-emulator evenals file transferprotocols.

Aldus besloot ik om dit programma te gebruiken en het warm water niet terug uit te vinden.

Het bewuste programma heette PCPLUS, maar eender welk programma met een goede terminal-emulator en Xmodem file transfer protocol is bruikbaar.

Zodoende moesten er maar 2 kleine machinetaal-programma's geschreven worden voor de Atom.

Deze heb ik als volgt genoemd : TTY65 en XMOD65 waar van het eerste de terminal-driver is en het tweede het file transfer-protocol voor zijn rekening neemt.

TTY65 verzet de input-/outputvectoren van de Atom, maar bij het verlaten worden deze terug geplaatst.

Verder gebruikt het ook nog een standaard aangesloten 6551 IC voor de RS232-conversie.

Van dit IC moet de interruptlijn aangesloten zijn aan de IRQ-ingang van de 6502-processor, daar deze lijn gebruikt wordt om het hangup-signaal en het break-signaal te detecteren.

Verder dient ook nog een VIA aanwezig te zijn in het systeem; deze is nodig voor de timing.

Schema's voor deze schakeling zijn reeds genoeg bekend, alleen moet er toch even aandacht aan de kabel geschonken worden.

```
Shield       1--------------1     Shield
RD           2--------------3     TD
TD           3--------------2     RD
RTS          4--+        +--4     RTS
CTS          5--|        |--5     CTS
DCD          8--+        +--8     DCD
DTR         20--------------6     DSR
DSR          6-------------20     DTR
Ground       7--------------7     Ground
```

De DSR-lijn wordt gebruikt om het hangup-signaal te detecteren.

De penaansluitingen zijn voor 25 pin-connectors waaraan aan de ene kant de overeenkomstige aansluitingen van het 6551 IC en langs de andere zijde de PC-connector aangesloten wordt.

Zoals u kunt zien, komt dat overeen met een NUL-modem-kabel. Alvorens de source-file TTY65 te assembleren moeten het adres van de 6551, de VIA en het startadres aan de source meegedeeld worden. Die source is redelijk leesbaar geschreven en dat kan dus moeilijk problemen scheppen.

Assembleer dat niet te hoog in het geheugen, liefst zo laag mogelijk zodat de driver niet in de weg zit met andere programma's. Hou er echter ook rekening mee dat het protocolprogramma hier achteraan geladen dient te worden. Met bufferplaats hoeft u geen rekening te houden, daar het filetransferprotocol hier zelf rekening mee houdt.

Na het opstarten van het programma met een *RUN %TTY65 moet het keyboard van de Atom doodgaan, behalve voor de Esc- en breaktoets.

De prompt moet ook verschijnen op de PC die het terminal-emulator-programma draait.

Indien de prompt terugkomt op het ATOM-scherm, dan is de DTR-lijn op de Pc niet hoog of is er een fout in het programma.

De DTR-lijn langs de kant van de PC moet geconfigureerd zijn zodat een hangup die lijn dropt.

Indien de prompt verschijnt, kunnen nu alle andere programma's gerund worden, zolang zij de driver maar niet overschrijven.
De terminal-emulator die gebruikt moet worden, is een Dump TTY of ANSI. De volgende controle characters worden herkend door de driver.
- CTRL S stopt tijdelijk een listing naar het scherm.
- CTRL Q herstart de listing, gestopt met CTRL S of door page-mode.
- CTRL N zet het scherm in page-mode; er worden maximum 22 lijnen op het scherm gelist; het listen stopt tijdelijk tot een CTRL Q gegeven wordt.
- CTRL O zet de page-mode listing af.
- ALT B, PCPLUS zendt een break naar de Atom; dat komt overeen met de breaktoets; alleen wordt niet de volledige break afgehandeld. Het is soms nodig 2 maal na elkaar ALT B te drukken alvorens dat commando herkend wordt.
- ALT H, PCPLUS zendt een hangup-commando naar de modem wat tot gevolg heeft dat de ATOM terugkeert in normale mode.
- ALT B en ALT H kunnen in andere terminal-emulators andere commando's inhouden; onthoud echter dat de ene een BREAK genereert en de andere een HANGUP.

Tot zover dit hoofdstuk over de terminal-driver.

Het volgende programma laat ons toe om files van en naar de ATOM te transporteren als de %TTY65 geladen is. Het gebruikte protocol is XMODEM checksum; daarvoor hoeft u echter niets te doen; XMODEM vindt dat zelf wel uit. Indien U echter in de source gaat kijken, zult u daar echter ook een subroutine vinden die voor CRC gebruikt kan worden; dat kan later geimplementeerd worden, doch op dit ogenblik is mij de juiste berekening voor die CRC nog niet bekend; dus als iemand kent, gelieve mij te contacteren.

Indien het juiste laadadres is ingesteld - ik laad hem normaal boven de terminal-emulator - kan XMOD65 geassembleerd worden. Let erop dat dit programma RANDOM-FILES gebruikt, zodat alleen de originele ATOM DOS of EDOS gebruikt kan worden.

Hoe werkt het Xmodem-protocol?

Laat ons aannemen dat we een file willen overzenden van de ATOM naar de PC. Aan de Prompt voeren we *RUN %XMODEM in en een nieuwe prompt verschijnt, die ons vraagt wat we willen uitvoeren. Hetzij downloaden (van de ATOM), uploaden (naar de Atom) of Exit. We kiezen hier D (alleen de eerste letter telt). De Atom gaat ons nu langs het PC-scherm om de filenaam vragen die we willen downloaden; we geven die in zoals we dat normaal zouden doen voor het ATOM DOS-commando. Bijv. %TTY65. De Atom gaat nu kijken of deze file bestaat. Als dat niet zo is, wordt dat gemeld en komen we terug bij de eerste prompt voor een nieuw commando. Indien die file bestaat, zal de ATOM dat meedelen met de desbetreffende boodschap. Het is nu tijd om het transfer te starten, daar de ATOM nu al staat te wachten op het download-programma van de PC; we selecteren bij PCPLUS Page Down, wat wil zeggen: Download naar de PC. De PC gaat nu de lokale filenaam vragen en het protocol starten. Als u echter te lang wacht, gaat de Atom in time-out en zal de hele sequentie aborteren. We geven de naam in en PCPLUS gaat nu verder met het XMODEM-protocol. Alle data die doorgestuurd worden, worden in blokken van 128 bytes overgestuurd; die blokken worden voorafgegaan door een SOH-character, het bloknummer en het bloknummer in 2 complement. Dan volgen 128 bytes data, gevolgd door een checksum-byte of 2 CRC-bytes.

De receiver zendt het "C" (43h) character en wacht op het eerste blok. Als dat niet komt binnen 7 seconden, wordt een nieuw "C" character verstuurd. Dit gebeurt 3 maal. Als de transmitter het "C" character samen met het eerste blok als antwoord terugstuurt, dan weet de receiver dat er zal gewerkt worden met CRC-blokcheck.
In de Atom werken we echter met checksum, zodat die 3 time-outs steeds optreden wanneer PCPLUS tracht te downloaden. De receiver zendt nu een "NACK" (15h) character en probeert dat ook 3 maal. Als daar ook niet op geantwoord wordt, dan veronderstelt de receiver dat er geen databloks komen en zendt een abort message "CAN" (18h) character.

Bijna dadelijk na het eerste "NACK" (15h) character antwoordt de transmitter met verzending van het "SOH" (01h) character, gevolgd door het bloknummer, het bloknummer in 2 complement, 128 bytes data en 1 checksum-byte. Vervolgens wacht de transmitter op het antwoord van de receiver. De receiver antwoordt met een "ACK" (06h) character als het blok data goed is aangekomen.
De receiver antwoordt met een "NACK" (15h) character als er zich een fout in het datablok bevond. Als er te veel geaccumuleerde fouten zijn (16), dan zal de receiver de transmissie aborteren met het "CAN" (18h) character. De transmitter zendt dan een "CR" (0Dh) om dat te beantwoorden.
Als de receiver het datablok geacknowledged heeft, zal de transmitter het volgend blok data versturen. Als er echter geen blok data meer te versturen is, zal de transmitter een "EOT" (04h) character sturen waarop de receiver zal antwoorden met een "CR" (0Dh) character.
Als er niet genoeg data meer aanwezig zijn om een blok te vullen, wordt dat blok opgevuld met het "SUB" (1Eh) character.
Als de Atom tracht te uploaden, zendt hij niet het "C" character maar dadelijk het "NACK" character, zodat de transfer dadelijk aanvangt.
De maximale snelheid die u met deze file-transfer kunt verwachten, ligt ongeveer op 5600 baud zelfs indien u de baudrate zou opdrijven tot 19200. Dat is niet aan de ATOM te wijten, maar aan de PC.

Wat heeft U nodig?

- Een PC met een terminal-programma en een XMODEM-file-transferprotocol.
- Een Atom met Atom DOS of EDOS, geen GDOS.
- Salfaa 2.0 of later.
- Een verbindingskabel.
- De volgende files:

TTY6551	De terminal driver.
XMODEM	Het Xmodem command file.
$XMODEM	Het declaratiefile voor Xmodem.
DXMODEM	Het download-gedeelte.
UXMODEM	Het upload-gedeelte.
SXMODEM	Allerlei subroutines in Xmodem gebruikt.

Hoe assembleren?

- Laad de source TTY6551 naar 2800h in het geheugen.
- Verander het laadadres van de machinecode zodanig dat het u past, maar dat het niet de buffers van de random-files overlapt.
- Verander ook het adres van het 6551 IC naar uw behoefte.
- Save die file terug naar disk.
- Run die file.
- Save die machinecodefile naar disk onder de naam %TTY65.
- Laad de commandfile XMODEM naar 6000h, pas het laadadres voor de machinecode aan zoals eerder, maar zorg er wel voor dat het %TTY65 niet overlapt wordt.

- Save deze file terug naar disk.
- Laad de declaratiefile $XMODEM naar 2800h, pas de adressen voor het IC 6551 en voor de VIA aan.
- Save deze file terug naar disk.
- Doe een *RUN XMODEM, het assembleren geschiedt nu vol automatisch.
- Save nu ook deze machinecode naar disk als %XMODEM.

Met dit en een beetje proberen moet het u toch lukken de communicatie met een PC tot stand te brengen.

Als het toch niet lukt, kunt u mij bereiken op de bijeenkomsten in de regio Limburg of bij mij thuis op het volgende adres :

Michel Van Leuven
Parsivalstraat 14
B-2580 ST.-KATELIJNE-WAVER
België
TEL.: 015/31.50.82
TEL. vanuit Ned.: 09-32-15-315082

S P S Sanders Print Service

Cassette interfaceprint	5.00
MDCR interface print	5.00
Battery backup printje	3.00
8K hoge geheugenprintje	5.00
#E000 naar #1000 voor oude schakelkaart	3.00
Omschakelprintje voor de 80K kaart	3.00
Bootstrapprintje de Moor	6.00
Voedingsprint MDCR 12V	5.00
Acoustische verbindingstester	5.50
Viaprint Z80 (Atombus)	15.00
8e printerbit	7.00

Al deze printen zijn ook gebouwd en getest te bestellen tegen kostprijs onderdelen en een symbolisch bedrag. Even bellen graag. (Tel. 04750-30401)

Bestellen: op de clubavond te Sittard of door overmaking van het bedrag (+1,-- per print verpakkings- en verzendkosten) onder vermelding van de print(en) op gironr: 794739 tnv: E. Sanders, Rosslag 13 te Herten.

De VIA of 6522.

In elk van onze ATOM's zit een zgn. VIA, aangesloten doorheen een buffertje naar de printerplug en eveneens aangesloten op de b-rij van plug P16.
De meesten van ons gebruiken het ding als printeruitgang en hebben zich nooit afgevraagd wat is een VIA en wat kan die zoal?

Toepassingen aangesloten op de b-rij zijn mij niet bekend, tenzij de oude EPROM programmer, zodat mag aangenomen worden dat de VIA een ondergewaardeerd broertje is.

VIA wil zeggen: Versatile Interface Adapter. Het is dus een chip die op soepele ma er moet toelaten te interfacen met externe toestellen, wat dat ook moge zijn.

De VIA is echter meer: hij bevat ook twee timers (tellers) die programmeerbaar zijn en waar ook heelwat mee kan gedaan worden.
Daarbuiten bevat hij ook een shift register, waarmede serie-parallel en parallel-serie conversie kan gedaan worden.

Omdat ik ooit mogelijkheden zocht voor externe aansluitingen, meer dan met de ingebouwde VIA, heb ik een tijd geleden een DUO VIA kaart ontworpen, waar dus twee VIA's op kunnen; dat was ook het moment om eens wat dieper in te gaan op de werking en interne samenhang. En daar schijnt dus op het moment wel wat interesse voor te bestaan zodat ik mij maar aan het schrijven heb gezet.
Heb echter voldoende consideratie met mij: ik ben zoals jullie ook een amateur!Het opstel valt uiteen in twee onderverdelingen: het hardware deel en het software deel. Er zijn natuurlijk verwijzingen van het ene naar het andere.

De HARDWARE.

De chip zit in een 40 pin package en bevat:
- Twee input/output poorten met bijbehorende controllijnen,
- Een shiftregister, programeerbaar.
- Twee interne timers/tellers die elk afzonderlijk programmeerbaar zijn.

Hij is dus een interface chip en zit dus op de verbinding tussen externe divices en de computer; aan de ene kant is er dus verbinding met de computer, aan de andere kant kan ik verbinding maken met andere toestellen.

De verbinding met de computer gaat langs:
- de 8 datalijnen die hangen aan de uitgangen van acht databuffers (bidirectioneel) en die in high-impedance blijven wanneer de chip niet geselecteerd is. De datatransfer, voor beide richtingen, gebeurt wanneer de clock hoog is.
- de clock-lijn, die dus het data-transport commandeert, maar ook van pas komt bij de timers en het shiftregister.
- de R/W lijn die de richting van het datatransfer bepaalt. - DDe IRQ- lijn die aangeeft dat de chip om een interrupt vraagt.Dit hele gebeureninterruptgebeuren is op zich al een interessant deel van de opbouw van de VIA.

- de reset die alle interne registers op nul zet.

Naar de komputer kant zijn er nog vijf lijnen: de chip-select die pos en neg uitgevoerd is, en de vier adesreslijnen waarmede de zestien interne registers kunnen bereikt worden.

In het ATOM-gebeuren wordt de chip-select geplaatst in het gebiedva van B800 tot BFFF, en wanneer men de vier laatste adreslijnen neemt voor de register-keuze, dan is het net alsof de VIA zestien opeenvolgende plaatsen inneemt in het geheugen.
LET ECHTER WEL OP: wat ik hiervoor verteld heb geldt niet voor de VIA die op ons fabrieksbord zit: die is geplaatst op B800 en de vijftien eropvolgende adressen; door de onvoldoende decodering kan dit echter ook bereikt worden op B900, BA00 en BB00!
Tot zover de verbindingen naar de komputer. De verbindingen naar de buitenwereld gebeuren langs twee poorten, de A en B poort elk 8 bit breed, en elk van deze poorten kan ik bit per bit ingang of uitgang declareren! Bovendien behoren bij elke poort een paar controllijnen (2) die weer in hun toepassing programmeerbaar zijn. In hun primaire toepassing kunnen ze gezien worden als een handshake-pair: maar ze kunnen dus meer.

De beide poorten zijn aan elkaar gelijk tot zover. De B poort is echter enigszins anders: de lijnen komen overeen met 1 TTL load en kunnen 1 TTL load driven; bovendien wordt bit 7 aangestuurd door de timers, en kan bit 6 dienen als ingang voor de timers.
De controllijnen van poort B kunnen ook ingezet worden als inputlijnen voor het shiftregister.

De acht-bit-breede poorten kunnen elk afzonderlijk in latch werking gezet worden: de data overdracht gebeurt dan op door de controllijnen bepaalde ogenblikken.

Het is duidelijk dat al deze instelbare mogelijkheden het werken met deze chip niet eenvoudig maken, maar toch: het is niet omdat er ingewikkelde dingen mee kunnen gebeuren, dat wij er ook gebruik van moeten maken: ook eenvoudige dingen kunnen en dat zullen we aantonen in het software gedeelte.

BESPREKING VAN DE ZESTIEN REGISTERS.

Zoals reeds gemeld worden met de vier adreslijnen zestien registers geselecteerd, elk met een eigen functie.
Geselecteerd met 0-0-0-0 wordt het poort B register, en naar gelang ik het als input of output heb gedefinieerd kan ik dan dit register lezen of schrijven.
Met 0-0-0-1 wordt het poort A register gekozen.

Met 0-0-1-0 wordt het data richtingsregister voor poort B gekozen, en met 0-0-1-1 hetzelfde register maar nu voor poort A.
Elke 0 in een bit van dit register maakt het overeenstemmende bit in de poort tot een INput; een 1 in een bit maakt het overeenstemmende bit in de poort tot een OUTput.

De zes volgende registers, vier voor timer 1 en twee voor timer 2, hebben te maken met het instellen van de timers. De timers w'rden het onderwerp van een afzonderlijk opstel.

Met 1-0-1-0 wordt het shift register gekozen.
Met 1-0-1-1 het Auxiliary control register,
met 1-1-0-0 het Periferal control register
met 1-1-0-1 het Interupt Flag register,
met 1-1-1-0 het Interupt Enable register, en
met 1-1-1-1 het poort A register, maar nu zonder dat dit de handshake lijnen beinvloedt.

DE DATA POORTEN.
Zoals reeds gezegd zijn er twee poorten A en B, die beiden bit voor bit tot ingang of uitgang kunnen gedeclareerd worden. Het is een goede gewoonte deze vastlegging te laten gebeuren bij het begin van een toepassing en deze tijdens de toepassing niet meer te veranderen. De definiering gebeurt in de data-direction registers:.een 0 maakt het overeenstemmende bit in de poort tot ingang,een 1 maakt het hvereenstemmende bit tot een uitgang.

Het eenvoudigste gebruik van de VIA: nml. een byte naar buiten sturen of een byte inlezen vraagt dus om twee acties: de poort definieren, naar de poort schrijven of van de poort lezen.

Een dergelijke transactie houdt geen rekening met het al of niet "klaar" zijn van het externe toestel om een byte te ontvangen of om er een te verzenden.M.a.w. het op elkaar synchonizeren ontbreekt; dit kan door te "handshaken" en dat gaat langs de controllijnen. De twee poorten onderscheiden zich echter als volgt: op de A po@rt kan Read en Write handshake, op de B poort kan alleen de Read handshake.

Normaal volgt de poort de data van de in of outputlijnen; het is echter mogelijk de poort te "latchen": zij houdt dan de gegevens van een bepaald ogenblik vast tot er weer eens gelatched wordt. Deze actie wordt gestuurd door controllijn 1 die een interruptflag zal setten: waarna een gepaste actie moet volgen. Of de poort gelatchd kan worden of niet wordt gedefinieerd in het ACRegister;
Hoe de controllijnen zijn gedefinieerd wordt opgeslagen in het PCRegister.
Wanneer we dus meer willen dan zomaar eens een byte outputten, dan moeten we ook de werking van de controllijnen vastleggen en dan is een goede kennis van de mogelijheden van zowel Auxiliary Controlregister als Peripheral Controlregister noodzakelijk.

PERIFERAL CONTROL REGISTER.
Het register is 8 bit "breed" en de 4 minst betekenisvolle bits besturen de controllijnen van de A poort, terwijl de 4 meest betekenisvolle bits de B poort definieeren.
Wat gezegd wordt voor de A poort geldt eveneens voor de B poort met uitzondering van de omstandigheid waarin het shiftregister ge selecteerd is: dan is de werking van deze lijnen anders.

Bit 0 bepaalt de werking van CA1 (controllijn 1 van poort A). Staat daar een 0 dan zal een negatieve sprong (van hoog naar laag) op de CA1 de interruptflag setten. Staat er een 1 dan zal een positieve sprong dezelfde actie tot gevolg hebben.
Bit 5 gedraagt zich net als bit 0, maar dan voor CB1.

De volgende drie bits bepalen de werking van CA2; is bit 3=0 dan is CA2 een inputlijn en zij zal een interrupt-flag in het interrupt register beinvloeden; is het bit 3 echter 1 dan is de lijn een

outputlijn; in beide "modes" bepalen bit 1 en 2 verschillende wijzen van werken.
Een samenhang van de werking vindt U in de bijgaande tabel uit "R6500 Hardware Manual" van Rockwell.De tweede tabel geeft de werking van de drie hoogste bits, van invloed op CB2, en de verschillen tussen beide zijn afleesbaar.

Welke van deze werkingswijzen zijn nu het meest voor de hand liggend in het gebruik? (de bespreking volgt voor poort A, voor poort B gelden gelijkaardige overwegingen.)
De handshake mode met bit 3,2 en 1= 1-0-0 ; Hierbij gaat CA2 laag telkens wanneer register A gelezen of geschreven wordt, en blijft dit tot er op CA1 een spanningssprong geregistreerd wordt: deze kan pos of neg zijn, naar gelang ingesteld door bit 0. De CA1 lijn moet dus door de randapparatuur gebruikt worden om mede te delen dat een byte werd ontvangen, of klaar staat om verzonden te worden.Op deze wijze "volgt" de komputer het rithme dat door de randapparatuur word aangegeven.
Is dit niet nodig, omdat de randapparatuur even snel of sneller is, dan kan gewerkt worden met 1-0-1 in bits 3,2 en 1: over CA2 wordt nu alleen maar een puls verzonden van een cyclus breed en volgend op het lezen of schrijven van het poortregister: op deze wijze wordt de randapparatuur ter kennis gegeven dat een byte klaar ataat.
Met 1-1-0 en 1-1-1 in de drie bereffende bits kan ik onder kontrole van het programma CA2 laag of hoog laten worden, en kan ik dus zelf de duurtijd van een "strobe" puls bepalen.
De control-lijnen kunnen dus zodanig gstuurd worden dat ze ofwel een bevel doorsturen, danwel een informatie binnenkrijgen; maar dan "weet" onze processor dat nog niet. Het doogeven van dergelijke informatie kan niet door deze lijnen te "lezen" of door ze ergens in geheugen op te slaan en deze informatie te gaan gebruiken. Om daar een oplossing aan te geven is er het interrupt register. Het heeft 7 bits die een of andere lijn of toestand bewaken en een achtste bit dat dan weer deze zeven "vlaggen" bewaakt.
Wat betekent dit?
Bit 0 van het interrupt register wordt gezet door een sprong op de CA2 lijn (pos of neg ngl. dat gedefinieerd werd) en wanneer tegelijkertijd het bit 0 van het interrupt enable register gezet is.
Op deze manier heeft elk bit een bepaalde bewaak-functie, met uitzondeing van bit 7 die de voorgaande zes bewaakt: zodra er een gezet is dan wordt deze ook gezet, en terzelfdertijd wordt een IRQ gegenereerd.
Elk gezet bit kan gereset worden door weer een ander voorval (bvb door poort A te lezen of te schrijven) OF door er een 0 in te schrijven.
Bit 7 echter kan alleen gereset worden door alle andere bits te resetten.
Het interrupt enable register wordt geset em gereset door erin te schrijven: DIT SCHRIJVEN is echter van een speciale soort:
Wanneer er in bit 7 een 1 staat dan wordt in de bits die ook een 1 hebben een 1 gezet; omgekeerd, staat in bit 7 een 0 dan komt in elk bit waar weer een 1 staat nu een 0. Raar maar zo is het. De inhoud van het IER veranderen zal dus meestal een dubbele actie vragen: eenmaal schrijven met bit 7 = 0 en eenmaal met bit 7 = 1.
Terug naar de werking met het interrupt register:
Wanneer een IRQ gegenereerd is en wanneer de interrupt toegelaten is (CLI) dan werkt de processor zijn lopend werk af, zet een aantal dingen op stack en springt dan naar het adres aangewezen door #FFFE en #FFFF; in de ATOM staat daar #FFB2 waar een kleine routine staat die onderzoekt of de onderbreking een BREAK was of een interrupt. Was het

een interrupt dan springt de processor naar de vector waar #204 naar wijst. In de niet uitgebreide ATOM staat daar #A000 en daar staat #40 in wat betekent RTI of return from interrupt; m.a.w. in de niet door de gebruikers uitgebreide ATOM gebeuren er geen interrupts, of: interrupt-routines moeten we zelf schrijven en het vector adres van deze routines moeten we dan in #204 zetten.
Het werken met interrupts vereist een goede kennis van de machinetaal en van de werking van de processor; voor het efficient werken met randapparatuur is het de aangewezen techniek.
We gaan hier niet dieper op deze techniek in maar geven wel een overzicht van wat zo een interrupt routine moet doen:
-kijken welk randapparaat de interrupt genereerde, en voor de VIA welk onderdeel;
-afhankelijk daarvan de erbij horende routines starten
-stoppen met terug te keren waar we gebleven waren.

SOFTWARE.
Om een idee te krijgen van de toepassingen van het besprokene volgen hierna enkele voorbeelden.

De printer-routine van de ATOM.
Deze begint op #FEFB; eerst wordt gekeken of de printer moet aangezet of afgezet worden, nml wanneer een P.$2 of $3 wordt gezonden. Het aanzetten, wat in ons geval wil zeggen de via klaar maken voor het verzenden van data, gebeurt op adres #FF27 en volgende. Hier moet opgemerkt dat in ATOM-WARE deel 3 enkele storende fouten zitten voor deze routine!
FF27 LDA @#7F ipv. F7
FF29 STA #B803 ipv. #B80C
Deze beide instructies maken de A poort voor de zeven laagste bits tot uitgang en voor het achtste bit tot ingang! Het achtste bit wordt namelijk door de verdere routine gecontroleerd om te weten of de printer het byte verwerkt heeft, of de printer dus "klaar" is om een byte te ontvangen.

De routine gaat verder met het instellen van het Periferal Control Register,en dit gaat met een constructie die de bestaande inhoud van dit register voor de B poort bewaart:
FF2C LDA B80C
FF2F AND #F0 bewaar de vier hoogste bits
FF31 ORA #0E maak bit 1,2, en 3 = 1
FF33 STA #B80C zet het resultaat in het PCR.
Het effect van deze actie is nu dat de lijn CA2 hoog staat: CA2 gaat dienst doen als STROBE signaal voor de printer.
Het "af" zetten van de printer gebeurt op een gelijkaardige wijze maar nu wordt het PCR in zijn oorspronkelijke stand gezet.
Tot zover is de Via in gereedheid gebracht om een byte naar de printer te zenden. Dit zenden gebeurt in de routine vanaf FF0F:
FF0F PLA haal het byte van de stack
FF10 BIT B801
FF13 BMI #FF10 deze twee instructies kijken of het achtste bit signaleert dat de printer nog bezig is (het BUSY-signaal) en dan wordt er gewacht; is de printer klaar dan volgt:
FF15 STA#B801 zet het byte op de poort,
FF18 PHA en ook terug op de stack
FF19 LDA #B80C
FF1C AND @#F0
FF1E ORA @#0C

```
FF20  STA #B80C
FF23  ORA @#02
FF24  BNE #FF33
..
FF33  STA #B80C
FF36  PLA
FF37  RTS
```

Deze bewerkingen rond #B80C zenden een negatieve puls op CA2, de STROBE, die aan de printer signaleert dat er een byte klaar staat.

De atom printerroutine is al bij al een zeer zuinige en ongecompliceerde oplossing, die zelfs erg simplistisch overkomt gezien in het licht van wat de via zoal meer kan. Bijvoorbeeld: het zenden van $2 en $3 zet de printer aan en af: maar in feite opent hij alleen het kanaal naar de printer zonder erop te letten of er wel een printer aangesloten is. Zet je de printer aan zonder dat er een aangsloten is dan hangt het systeem: nochthans heeft de via een timer waarmede een time-out zou kunnen ingeschakeld worden! De handshaking is helemaal software gestuurd, waar de via de hardware aan boord heeft om dat zelf te doen! Een en ander zal wel te maken hebben met het feit dat men zuinig moest zijn met de beschikbare memorie!

Een tweede voorbeeld is de printerroutine die ikzelf geschreven heb om langs de via een bolkopschrijfmachine aan te sturen. Daarbij stelden zich een hoop problemen: ten eerste accepteert deze schrijfmachine geen ascii-code maar selectric-code: er moet dus geconverteerd worden, ten tweede geldt de selectric code alleen voor letters en tekens en niet voor stuursignalen zoals spatie,nieuwe regel, tabulator, cariage return en dgl., verder moeten een aantal signalen van de schrijfmachine doorgegeven worden naar de computer: een combinatie van busy toestanden voor de cariage return, de machinebevelen, de letterbevelen, de stand van de printkop i.v.m. hoofd- of kleine letters.
Een en ander deed mij besluiten om de beide printerpoorten te gebruiken: de ene als ingangspoort om al die feed-back signalen te lezen, de andere als uitgangspoort waarlangs de selectric signalen, gemengd met machine-informatie verstuurd kan worden.

Om de geschiedenis te kunnen volgen is het in salfaa geschreven source programma "drukker" bijgevoegd.
Lijn 120 tot 210 definieert de gebruikte via-register adressen: zoals u ziet zijn dit er nu niet 3 maar 9.Lijn 260 tot 530 genereert een nieuwe printroutine in plaats van de originele atom routine op #FE52, om te kunnen springen naar mijn bolkop-aansturing.(op #208 staat in mijn systeem nu #BB80, het startadres van de nieuwe routine, ipv FE53)
In lijnen 630 tot 960 wordt het in en uitschakelen gedaan, dwz. het initialiseren van de registers en het kijken of de printer wel is ingeschakeld, is dit niet dan volgt een dringend bevel om dit alsnog te doen. In deze routine zit een subroutine ms1 genaamd: in deze wordt met behulp van de timer 1 ms gewacht, vemenigvuldigd met de inhoud van het Y register op het moment dat naar de subroutine gesprongen wordt: ik kan dus wachttijden van 1 tot 265 milliseconden inbouwen. In tegenstelling tot de originele atom printerroutine worden bovendien bij het verlaten alle registers terug in hun beginstand gezet.
Van 970 tot 1370 wordt de ascii kode omgezet tot code die voor mijn bolkop-schrijfmachine nuttig is: eerst worden de machine bevelen uitgefilterd en eventueel doorgestuurd, dan wordt omgezet in

selectriccode. Voor deze wordt opgestuurd wordt eerst gekeken of het gaat om hoofd of kleine letters, en of de kop goed gedraaid staat: het is namelijk zo dat niet telkens teruggedraaid wordt naar een neutrale stand: er wordt alleen gedraaid wanneer de toestand van een volgende letter anders is dan die van de voorgaande.
Moet er gedraaid worden dan word dit eerst gedaan, gewacht tot het gedaan is en dan eerst wordt de lettercode doorgegeven. Het bekrachtigen van al de verschillende relais vergde verschillende bekrachtigingstijden die nogal scherp luisterden: was deze te kort dan volgde geen actie, te lang dan werd de actie gerepeteerd; daarom wordt veelvuldig beroep gedaan op de wachtroutine met verschillende wachttijden.
Vanaf 1660 staan de gebruikte subroutines, waarvan ms1 de interessantste is omdat hij de werking van timer 1 illustreert: het ACR wordt geinitialiseerd, de telregisters van de timer die start bij het schrijven in het high-byte; komt de timer op 0 dan wordt een overeenstemmend bit geset in het IFR om de wachttijd te stoppen.

Vanaf 2000 staat de tabel die de conversie naar selectric-code doet, en vanaf 2200 staan de relais-tijden.

In een volgend artikel zal nader ingegaan worden op het gebruik van de timers, het interrupt register en het schuifregister.

Vermits zovelen van ons nog met andere elektronica bezig zijn is de via het aangewezen ding om deze elktronica te bedienen en te sturen door onze ATOM en ik sta er eigenlijk wat over verbaasd dat nog zo weinig in deze richting gedaan werd. Ofwel werd het nooit openbaar gemaakt en dan is dit een oproep: kruip in de pen!

B E R I C H T van de F E D E R A T I E .

Op 29 September a.s. vindt de jaarvergadering van de Federatie plaats, waarbij de diverse regio-bestuurders de koers van de Federatie bepalen.
De agenda en andere bescheiden zal hen nog worden toegezonden.

Een van de agendapunten is het vaststellen van de 2e landdag 1990.
De leden van de onderscheiden regio's zouden wij ten overvloede willen vragen hun ideeen en verlangens ten aanzien van alle clubactiviteiten voor die datum te willen deponeren bij het desbetreffende regio-bestuur , danwel bij de secretaris of penningmeester van de federatie.

```
>L.
   10 PROGRAM DRUKKER
   20REM THU.25-08-87
   30 ASM-V
   40 PASS2;P."PASS1";GOSUBa
   50 PASS1;P."PASS2";GOSUBa
   60 ASM-U
   70 ASM-L
   80 END
   90aASM-B
  100 .opt:0100 0000
  110:base          =#7B80
  120:via           =#B010
  130:drb           =via+#0
  140:dra           =via+#1
  150:ddrb          =via+#2
  160:ddra          =via+#3
  170:t1llb         =via+#4
  180:t1lhb         =via+#5
  190:acr           =via+#0B
  200:pcr           =via+#0C
  210:ifr           =via+#0D
  220:text          =#F7D1
  230:toets         =#FE94
  240 .code base
  250 .ram  #7B80
  260:begin                \NU IS DIT #FE52
  270 PHP                  \ALLEEN JSR drukker IS ANDERS
  280 PHA
  290 CLD
  300 STY#E5
  310 STX#E4
  320 JSR drukker          \IS ALS BEST.A TOM ROUTINE
  330 JSR#FCEA
  340 PLA
  350 LDX#E4
  360 LDY#E5
  370 PLP
  380 RTS
  390:drukker              \NIEUWE ROUTINE
  400 PHA
  410 CMP@#02              \AANZETTEN?
  420 BEQ zetaan
  430 CMP@#03
  440 BEQ zetaf
  450 CMP #FE
  460 BEQ drukniet
  470 LDA drb              \STAAT PRINTER AAN?
  480 AND@#10
  490 BNE drukniet
  500 LDApcr
  510 AND@#0E
  520 BEQ drukniet
  530 JMP drukibm
  540:zetaan
  550 LDA@#FF              \ZET POORTEN
  560 STA ddra             \A
  570 STA dra
  580 LDA ddrb             \EN OOK B
```

```
590 AND@#0F
600 STA ddrb
610 LDA pcr            \ZET CA2=1
620 AND@#F0
630 ORA@#0E            \EN BEWAAR ANDE RE BITS VAN pcr
640 STA pcr
650 LDA@#00
660 STA#FE
670:test               \OF IBM INGESCHAKELD
680 LDY tijd3
690:chk2
700 JSR ms1
710 DEY
720 BNE chk2
730 LDA drb            \IS BIT4 =0?
740 AND@#10            \ZONIET
750 BEQdrukniet
760:meded
770 JSRtext
780.ASCI "ZET PRINTER AAN"
790.BYTE #07,#0D,#0A
800 NOP
810 JMP test
820:zetaf
830 LDA@#00            \RESET POORT A
840 STA ddra
850\LDA ddrb
860\AND@#0F
870\ORA@#00
880 STA ddrb
890\AND@#F0
900\ORA@#0
910 STA pcr
920 LDA@#0A
930 STA#FE
940:drukniet
950 PLA
960 RTS
970:drukibm            \IBM PRINT ROUTINE
980 JSR klaar
990 PLA
1000 PHA
1010 CMP@#08           \IS HET BS?
1020 BNE htab
1030 LDA@#DF;PHA
1040 LDX tijd1
1050 JMP mach
1060:htab
1070 CMP@#09           \IS HET HTAB?
1080 BNE lf
1090 LDA@#F7;PHA
1100 LDX tijd2
1110 JMP mach
1120:lf
1130 CMP@#0A           \IS HET LF?
1140 BNE cr
1150 LDA@#FD;PHA
1160 LDX tijd1
1170 JMP mach
```

```
1180:cr
1190 CMP@#0D            \IS HET CR?
1200 BNE space
1210 LDA@#FB;PHA
1220 LDX tijd2
1230 JMP mach
1240:space
1250 CMP@#20            \IS HET SPACE?
1260 BNE zoek
1270 LDA@#BF;PHA
1280 LDX tijd1
1290 JMP mach
1300:zoek
1310 SEC
1320 SBC@#21            \MAAK VAN ASCII-CODE x INDEX
1330 BCC drukniet       \GEEN LETTER
1340 CMP@#60
1350 BCS drukniet       \GEEN LETTER
1360 TAX                \MAAK VAN ASCII-CODE x INDEX
1370 LDA tabel1,X       \HAAL IBM CODE
1380 PHA
1390 LDXtijd1
1400:shift
1410 EOR drb            \VGL M SHIFTSTATUS:
1420 AND@#80            \NEEM BIT 7
1430 BEQ mach
1440 LDA@#7F
1450 STA dra            \SHIFTREL BEKR.
1460 JSRpuls
1470 JSR klaar
1480 LDY@#A0
1490:chk3
1500 JSR ms1
1510 DEY
1520 BNEchk3
1530:mach
1540 PLA                \HAAL CODE VAN STACK
1550 ORA@#80            \MAAK BIT 7=1
1560 STA dra            \STUUR LETTER RELAIS
1570 JSRpuls
1580 LDA@#FF            \CLEAR LETTER RELAIS
1590 STA dra
1600:wacht
1610 JSR ms1
1620 DEX
1630 BNEwacht
1640 PLA
1650 RTS
1660:puls               \BEKRACHTIG RELAIS
1670 LDApcr
1680 AND@#F0            \MAAK CA2=0
1690 ORA@#0C
1700 STA pcr
1710 JSR ms20
1720 LDA pcr
1730 AND@#F0
1740 ORA@#0E
1750 STA pcr
1760 RTS
1770:ms20
```

```
1780 LDY tijd4
1790:chk
1800 JSRms1
1810 DEY
1820 BNEchk
1830 RTS
1840:ms1
1850 LDA@#00;STAacr
1860 LDA@#A0;STAt1llb
1870 LDA@#00;STAt1lhb
1880 LDA@#40
1890:chk1
1900 BIT ifr
1910 BEQ chk1
1920 LDA t1llb
1930 RTS
1940:klaar
1950 LDAdrb
1960 AND@#60
1970 EOR@#60
1980 BNE klaar
1990 RTS
2000:tabel1
2010.BYTE #B0,#90,#F0,#F2,#2E\0
2020.BYTE #80,#A2,#E0,#DC,#84
2030.BYTE #48,#9E,#FE,#58,#68\1
2040.BYTE #62,#00,#50,#10,#22
2050.BYTE #60,#70,#20,#30,#72\2
2060.BYTE #E8,#D8,#E0,#C8,#DC
2070.BYTE #3C,#E2,#38,#76,#34\3
2080.BYTE #24,#64,#1C,#0C,#66
2090.BYTE #78,#4C,#74,#26,#08\4
2100.BYTE #54,#2A,#6C,#7C,#28
2110.BYTE #6A,#44,#14,#18,#7A\5
2120.BYTE #04,#6E,#40,#E0,#A0
2130.BYTE #DC,#2C,#7E,#D0,#B8\6
2140.BYTE #F6,#B4,#A4,#E4,#9C
2150.BYTE #8C,#E6,#F8,#CC,#F4\7
2160.BYTE #A6,#88,#D4,#AA,#EC
2170.BYTE #FC,#A8,#EA,#C4,#94\8
2180.BYTE #98,#FA,#84,#EE,#C0
2190.BYTE #E0,#AE,#DC,#6F,#DF\9
2200:tijd1              \MACHIEN
2210.BYTE#48
2220:tijd2              \LONGFU
2230.BYTE#96
2240:tijd3              \INSCH.
2250.BYTE#16
2260:tijd4              \PULSD.
2270.BYTE#40
2280:end
2290.END
2300 RETURN
```

NIEUW LEVEN IN DE ATOM - BROUWERIJ ? .

Op sterven na dood, als je zo wel eens kijkt naar de omvang en inhoud van de artikelen in ons aller ATOM-nieuws.
Ogenschijnlijk, want steeds opnieuw blijkt er toch nog aan de ATOM gesleuteld te kunnen worden, waarbij het langzamerhand natuurlijk veel meer gaat om de vraag hoe je een probleem oplost, en nog slechts in beperkte mate directe practische toepassingen aan de orde zijn.
Dat daarbij de oorspronkelijke opzet en de modificaties door allerlei clubleden en commissies soms wel eens hinderlijk kunnen zijn , is ons allen wel bekend.
Met name in het gebied #Bxxx is er zo het een en ander dat in dit opzicht nog wel eens parten speelt.
Of het nu gaat om meer uitgebreide toepassingen van de ATOM , of om een mini-besturings-ATOM ; niet zelden loop je aan tegen het probleem van externe communicatie en te weinig geheugen.

Dit probleem , door Henk Bastings op tafel gebracht , leek de moeite waard , zoals jullie in het vorige regionieuws hebben kunnen lezen, op een clubavond in Juli te bespreken.
Als uitgangspunt werd daarbij gebruikt een globaal grafisch overzicht van de bezetting van het #Bxxx-gebied (zie Bijlage 1).

1. Het als eerste, reeds lang bekende, probleem moet worden afgerekend met de al te overdadige beslaglegging van adressen voor selectie van de 8255 en de 6522.
Dit kan weliswaar op een eenvoudige wijze geschieden door additionele IC's in te schakelen (eerder al gepubliceerd), in het voorstel van Bastings zou dit gecombineerd kunnen worden met overige decodering , d.m.v. een PAL.

2. Het tweede deel van het voorstel heeft betrekking op de gedachte, equivalent aan de toewijzing van gebieden voor beeld, geluid en fdc, het #B8xx-gebied aan te wijzen voor alle externe communicatie.
Dit zou betekenen, dat eventueel reeds elders geplaatste RS232-verbindingen hier naar toe zouden moeten worden omgezet.
Ook de op PL8 aangesloten externe verbindingen zouden daaronder vallen.

3. Het derde deel van het voorstel heeft betrekking op het gebruik van het aldus vrijgekomen gebied.
Met gebruik van de vandaag verkrijgbare grote geheugen-IC's en eproms is het mogelijk alle voor het #Bxxx-gebied benodigde software, op te slaan in het gebied #B100 tot #B7FF op de wijze zoals dat gebeurt op de "schakelkaart", dus parallel aan elkaar op hetzelfde adres.

Ook de daarvoor benodigde "schakelsoft" zou in dit gebied moeten liggen.
Wordt nu een bepaald in - of uitvoerorgaan gekozen, dan wordt tevens de daarvoor benodigde software geselecteerd.

Dit heeft tot gevolg , dat "de normale geheugen-gebieden " (#1000 #7FFF en #8200/#9800-#AFFF) geheel voor programma's en data ter beschikking komen.

In de daaropvolgende discussie kwam aan de orde :

* Is zo'n diepgaande wijziging van de ATOM nu nog wel zinvol.

Vanuit het standpunt van de eenvoudige gebruiker , uiteraard niet; maar , de ATOM heeft alleen daarom zoveel furore gemaakt , omdat er zo heerlijk aan gesleuteld kan worden en vanuit die kant is er gewoon de uitdaging , om met moderne en goedkope middelen , een oplossing voor dit probleem te vinden.
Bovendien , maar dat is al in het begin gezegd, zal een meer algemene toepassing van de ATOM alleen maar mogelijk zijn door meer en meer algemene externe communicatie.

* Gebruik van een PAL heeft welliswaar veel voordelen, doordat er met een IC een groot aantal functies van allerlei aard gemaakt kunnen worden, nadeel is dat het programmeren speciale apparatuur vereist, en de programmering is " vast ", dus niet even een adres-sering veranderen door het omsolderen van een draadje.

Daar staat tegenover , dat er op dit moment voldoende mogelijkheden zijn ,om deze PAL's met de gewenste inhoud, op een redelijk snelle en redelijk goedkope manier aan te maken.

* Voor diverse I/O-apparaten is de software uitsluitend aanwezig in de vorm van machine-taal, de sources en documentatie ontbreekt.
Dat betekend dat aan het omzetten van deze software naar het #Bxxx-gebied nog wel eens de nodige moeilijkheden en in ieder geval het nodige werk verbonden zal zijn.
Met behulp van de momenteel beschikbare source-maker-programmatuur zou dit echter toch tot de reeele mogelijkheden behoren.

Ter vergadering werden enkele voorontwerpen voor een deel van de benodigde hardware getoond.

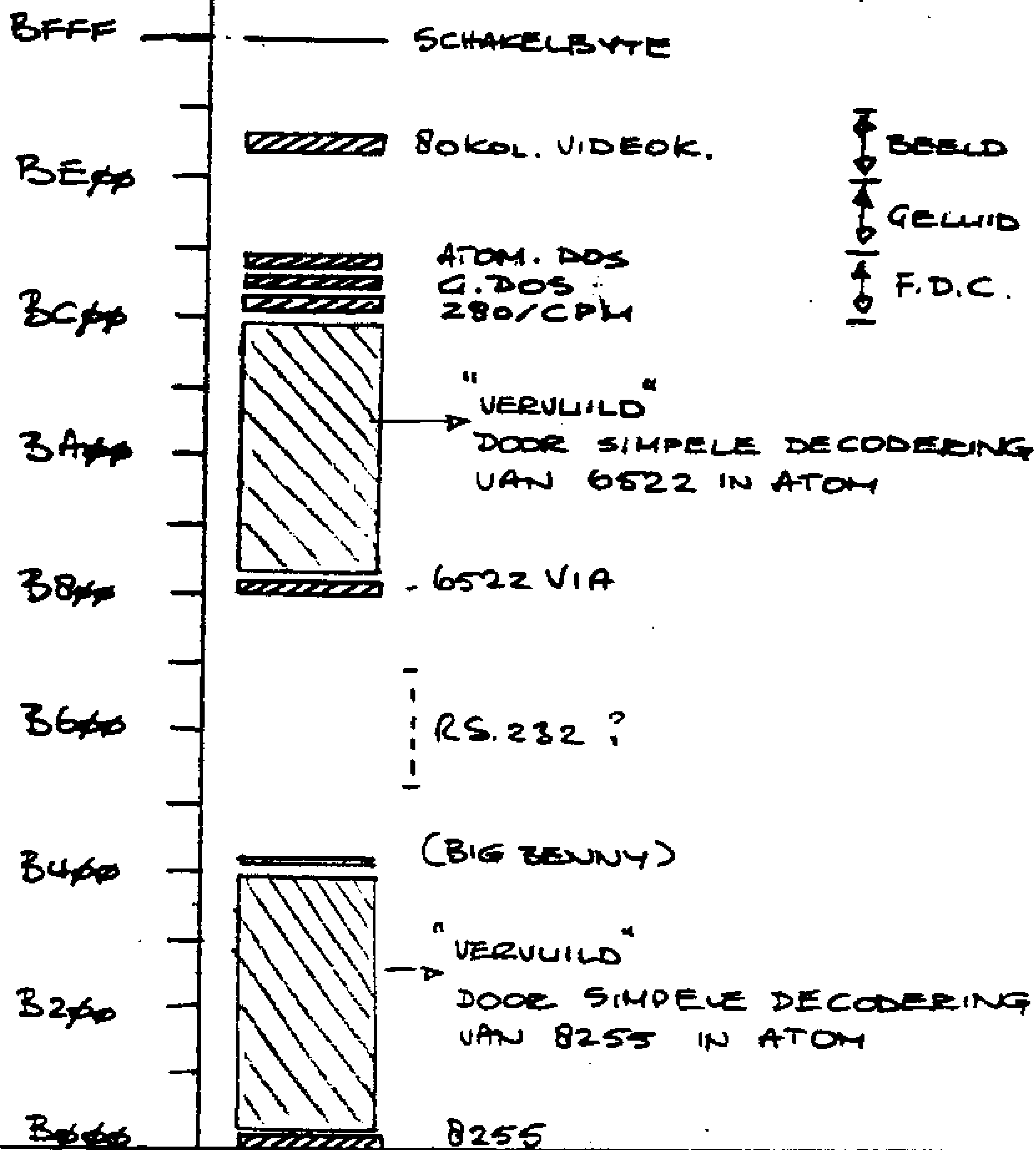
Conclusie van de aanwezigen was:
dat voor diegenen ,die echt in ontwikkelings-werk als zodanig zijn geinterresseerd, aantrekkelijke mogelijkheden aanwezig zijn;
Het is de moeite waard op de ingeslagen weg voort te gaan.
De volgende clubavond zal de discussie worden voortgezet.

Uw correspondent ;
Bruno Tossaint.

HUIDIGE BEZETTING

#Bxxx - GEBIED

AFSPRAKEN



NIEUWE SITUATIE

#3xxx - GEBIED

AFSPRAKEN



3FFF — SCHAKELBYTE

80 KOL. VIDEO K. — BEELD

3E00 — GELUID

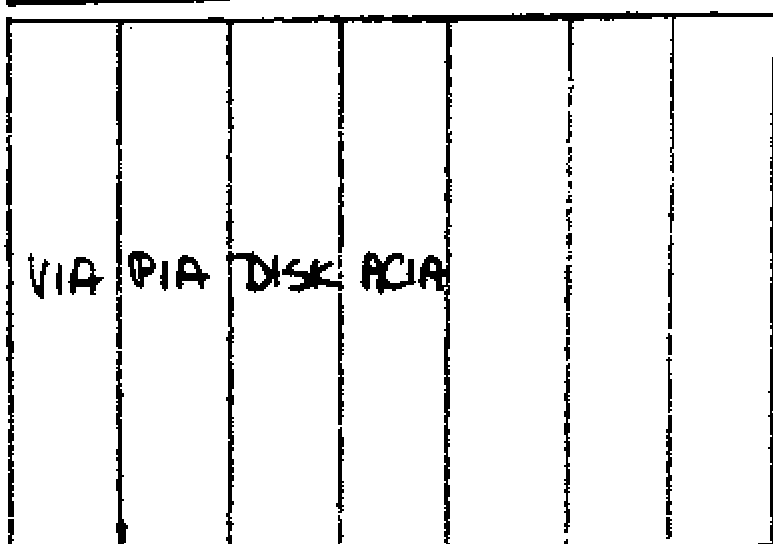
ATOM. DOS
G. DOS — F.D.C
Z80 / CPM

3C00

3A00

6522 VIA — ALLE EXTERNE I/O.

3800



DIVERSE
"DRIVERS"/SOFTW.
PARALLEL

3600

3400

SCHAKEL-
SOFT

3200

3000 — 8255

GEBRUIK UW ELECTRON EENS NUTTIG ...

PRINTER-BUFFER VOOR DE ATOM

door Roland Leurs

Veel printers hebben intern een buffer zitten die een hoeveelheid tekst op kan slaan, zodat de Atom niet al te lang hoeft te wachten totdat de tekst naar de printer verstuurd is. Maar deze buffer is meestal maar iets van 2 tot 4 kb groot en dat is niet genoeg voor een lange tekst. Dus toch maar wachten.

Een losse printerbuffer van minstens 32 k is voor een Atomist eigenlijk te duur, want zo groot zijn onze verhalen nou ook weer niet. Slimme programmeurs losten deze problemen op door een printerbufferprogramma te maken dat een stuk van het Atom geheugen gebruikt als printerbuffer. Maar ik blijf zeuren: als ik een buffer maak van 16 kbytes, hoe past dan een EDIT80 file van 20 k nog erbij ?

Dan toch maar een externe buffer maken, maar hoe? Als je nadenkt wat een buffer precies bevat is het antwoord simpel: geheugen + besturing van binnenkomende data en uitgaande data. Dat is dus een klusje van niets voor een Electron. En zie hier, het programma voor een printerbuffer van bijna 28 kbytes, en dat alles voor de prijs van een kabel tussen Atom printerpoort naar Electron VIA print. Verder moet de Electron ook een printeruitgang hebben (is toch wel logisch).

Bij dit alles is er een kleine maar: de Electron zit intern ook nog wat te rotzooien met een printerbuffer van 128 bytes en daardoor zijn er soms problemen met de timing en zo. Kortom, soms mist de Electron enkele karakters. Dit probleem heb ik opgelost door de Atom te vertragen. Dit is een kort machinetaal programma dat u moet aanroepen voordat u gaat printen. De Atom wacht dan even als een karakter verstuurd is.

Mocht u geen Electron hebben die aan bovenstaande eisen voldoet dan kunt u het programma aanpassen voor op uw tweede Atom te laten lopen. Of u past het programma aan zodat u de Atom als printerbuffer voor uw PC gebruikt. Alles kan en voor het gemak heb ik de source uitbundig van enig commentaar voorzien en u krijgt er zelfs een Nassi-Schneider diagram bij, zodat u kunt zien waar wat gebeurd.

Nog even de werking van de buffer uitleggen. De Electron kijkt of er een karakter aankomt (dmv strobe signaal). Dit karakter wordt dan ingelezen en opgeslagen in de buffer. Vervolgens wordt gekeken of de printer ready is. Zo ja, dan wordt een karakter uit de buffer naar de printer gestuurd. Als de printer niet klaar is, wordt steeds de bufferpointer verhoogd, totdat deze helemaal rond is. Dan zijn de bufferpointer en de printpointer gelijk aan elkaar. De buffer is dan vol en de capslock led van de Electron gaat uit. Vanaf dit moment blijven de twee pointers elkaar achterna lopen. Stopt de Atom met printen, dan kan de printer dus nog 28 kbytes lang doorgaan. Maar bij een tekstfile van

#2800-#7FFF raakt de buffer net niet vol dus we hoeven over het algemeen niet lang te wachten voordat we de Atom weer tot onze beschikking hebben. Overigens, tijdens het inlezen in de buffer kan het Electron beeldscherm ook troep gaan vertonen omdat ik deze 8 kb (in mode 7) ook nog bij de buffer heb genomen. Geen reden tot paniek dus!

Al met al een slordig verhaal (want ik heb er bijna geen zin meer in) van een mooi en nuttig programma. Bekijk de listing maar eens en doe er iets nuttigs mee!

```
 10 REM ELECTRON ALS PRINTERBUFFER
 20 REM GESCHREVEN DOOR ROLAND LEURS
 30 REM
 40 REM LET OP : ASSEMBLERCODE KOMT OP &F00 ! ! ! ! !
 50 REM
 60
 70 VIA      = &FD00          : REM VIA 1
 80 PRB      = VIA+&0
 90 DDRB     = VIA+&2
100 PCR      = VIA+&C
110 FIR      = VIA+&D
120 IER      = VIA+&E
130
140 P        = &90            : REM PRINT POINTER
150 B        = &92            : REM BUFFER POINTER
160 VOL      = &94            : REM BUFFER STATUS
170 BEGIN    = &1000          : REM BEGINADRES VAN DE BUFFERRUIMTE
180 EIND     = &7DFF          : REM EINDADRES VAN DE BUFFERRUIMTE
190 WRCH     = &FFEE
200
210 FOR PASS%=0 TO 3 STEP 3
220
230 P%=&F00                   : REM LET OP     <====
240
250 [ OPT PASS%
260 .buffer    LDA #&00              B-POORT VIA WORDT INPUT
270            STA DDRB
280            LDA #&7F              DISABLE ALLE INTERRUPTS VAN VIA
290            STA IER
300            LDA #&C0              INSTELLING CB1 & CB2
310            STA PCR
320            LDY #&00              LET Y = 0 , ivm indirecte adressering
330            LDA #BEGIN DIV 256    INIT POINTERS
340            STA P+1
350            STA B+1
360            LDA #BEGIN AND 255
370            STA P
380            STA B
390            LDA #FALSE            LET VOL = FALSE
400            STA VOL
410            LDA #&02              ZET PRINTER AAN
420            JSR WRCH
430 .repeat    LDA #&E0              LET BUSY = TRUE
440            STA PCR
450            LDA VOL               IF VOL = FALSE
460            BNE ready
470            LDA FIR                 EN STROBE = TRUE
```

```
 480          AND #&10
 490          BEQ ready
 500          LDA #&E0                THEN LET BUSY = TRUE
 510          STA PCR
  520             LDA PRB                     LEES DATA (strobe wordt
                                                        nu ook
gereset!)
 530          STA (B),Y               BERG DATA OP IN DE BUFFER
 540          INC B                   VERHOOG BUFFER POINTER
 550          BNE L1
 560          INC B+1
 570 .L1      LDA B+1                  IF B = EINDE GEHEUGEN
 580          CMP #EIND DIV 256
 590          BNE L2
 600          LDA B
 610          CMP #EIND AND 255
 620          BNE L2
 630          LDA #BEGIN DIV 256       THEN LET B = BEGIN GEHEUGEN
 640          STA B+1
 650          LDA #BEGIN AND 255
 660          STA B
 670 .L2      LDA B                    IF B = P
 680          CMP P
 690          BNE ready
 700          LDA B+1
 710          CMP P+1
 720          BNE ready
 730          LDA #TRUE AND 255        THEN LET VOL = TRUE
 740          STA VOL
 750          LDA #&30                 MAAK CAPS LK LED UIT (buffer vol)
 760          STA &FE07
 770 .ready   LDA &FC72                IF PRINTER IS READY
 780          BMI NIETPRT
 790          LDA B                     EN B <> P
 800          CMP P
 810          BNE print
 820          LDA B+1
 830          CMP P+1
 840          BNE print
 850          LDA VOL                   OF VOL = TRUE
 860          BEQ NIETPRT
 870 .print   LDA (P),Y                THEN LEES DATA UIT BUFFER
 880          PHA                      BEWAAR DATA EVEN OP STACK
 890          LDA #&01                 ZET PRINTER AAN VOOR 1 KARAKTER
 900          JSR WRCH
 910          PLA                      HAAL DATA WEER TERUG VAN STACK
 920          JSR WRCH                 STUUR DE DATA NAAR DE PRINTER
 930          INC P                    VERHOOG PRINT POINTER
 940          BNE L3
 950          INC P+1
 960 .L3      LDA P+1                  IF P = EIND VAN GEHEUGEN
 970          CMP #EIND DIV 256
 980          BNE L4
 990          LDA P
1000          CMP #EIND AND 255
1010          BNE L4
1020          LDA #BEGIN DIV 256       THEN LET P = BEGIN VAN GEHEUGEN
1030          STA P+1
1040          LDA #BEGIN AND 255
```

```
1050            STA P
1060 .L4        LDA #FALSE              LET VOL = FALSE
1070            STA VOL
1080            LDA #&B0                MAAK CAPS LK LED AAN (buffer
                                                              niet vol)
1090            STA &FE07
1100            BNE L5
1110 .NIETPRT   LDA VOL                 IF VOL = TRUE
1120            BNE until               THEN LET BUSY FALSE
1130 .L5        LDA #&C0                LET BUSY = FALSE
1140            STA PCR
1150 .until     JMP repeat              UNTIL BREAK WORDT INGEDRUKT
1160 ]
1170 NEXT PASS%
1180 END
```

Zoals u ziet lijkt deze assembler veel op SALFAA of minias dus het aanpassen geeft voor de Atom weinig problemen. De code past binnen 256 bytes.
Hieronder nog even de listing van het vertragingsprogramma voor op de Atom.

```
 10 REM LANGZAME WRCH
 20 REM VOOR PRINTER BUFFER
 30 REM DOOR ROLAND LEURS
 40 DIM TT4;FOR I=0 TO 3;TTI=#FFFF;NEXT
 50 FOR I=0 TO 2;P=#2600; REM ZELF UW ADRES INVULLEN   <========
 60[:TT0;LDA #209;CMP @(TT2/256);BEQ TT1
 70 LDA #209;STA TT3+2;LDA #208;STA TT3+1
 80 LDA @(TT2/256);STA #209;LDA @(TT2%256);STA #208
 90:TT1;RTS
100:TT2;PHA                        \SAVE ACCU
110 LDA @#C0                       \LAADT INDEX
120 STA TT4
130 NOP                            \WACHT
140 DEC  TT4                       \VERLAAG INDEX
150 BNE P-4                        \NOG NIET 0, DAN NOG EEN KEERTJE
160 PLA                            \ACCU TERUGHALEN
170 JMP TT3                        \GA MAAR AFDRUKKEN
180:TT3;JMP #FFFF
190:TT4 BRK
200] NEXT I
210 PRINT "AANROEPEN MET LINK #"&TT0'
220 END
```

Deze routine dus voor het printen aanroepen (vanuit tekstverwerker mbv *GO 2600). Verder natuurlijk de vriendelijke groeten van Roland.

NS - DIAGRAM BIJ ELECTRON ALS PRINTERBUFFER

INIT VLAGGEN EN CONSTANTEN

VIA B-POORT := INPUT

DISABLE INTERRUPTS VAN DE VIA

CB1 := INGANG, ACTIEF BIJ

CB2 := UITGANG

INIT DE POINTERS P (printer) EN B (buffer)

REPEAT

- BUSY := TRUE
- STROBE ACTIEF & BUFFER NIET VOL
  - N: —
  - J:
    - LEES DATA IN
    - BERG OP IN BUFFER
    - B := B+1
    - B = TOP
      - N: —
      - J: B := BEGIN
    - B = P
      - N: —
      - J: VOL := TRUE
- (IS PRINTER READY) & (B<>P OR VOL=TRUE)
  - N:
    - —
    - VOL = TRUE
      - N: BUSY := FALSE
      - J: —
  - J:
    - LEES DATA UIT BUFFER
    - PRINT DATA
    - P := P+1
    - p = TOP
      - N: —
      - J: P := BEGIN
    - VOL := FALSE
    - BUSY := FALSE
    - WACHT EVEN

UNTIL RESET

Maak zelf uw eigen modelbaan-besturing met uw Atom

door Roland Leurs

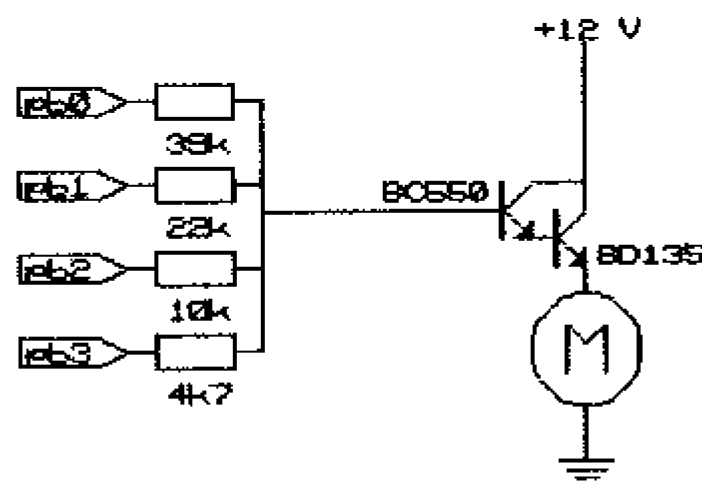
Een van de zaken waar een Atom en een Electron uitermate voor geschikt zijn is het I/O gebeuren. In onze regio zijn we al uitgebreide thema-avonden aan het houden omtrent dit onderwerp.

In juli ontving ik van de heer W. v. Norden een brief waarin hij het besturen van een modelbaan ook even vermeldde. Ik heb het gevoel dat er meedere leden in onze club zijn die hier ook wel eens wat aan willen doen dus schreef ik het nu volgende artikel over treintjes.

Het interface

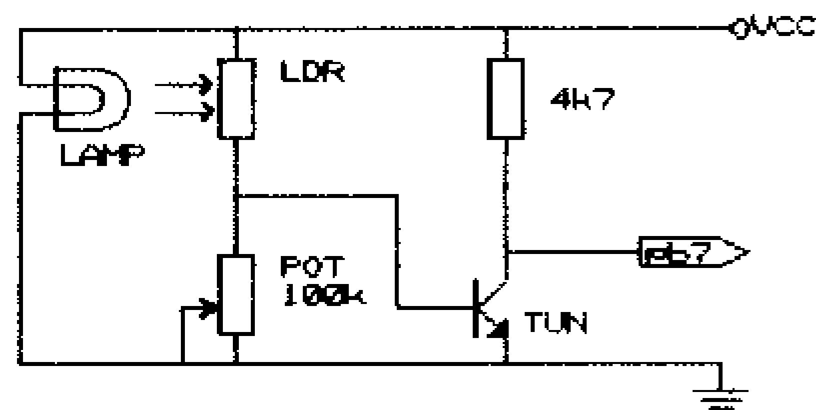
Om een trein te laten rijden en laten stoppen is in principe niet veel hardware nodig. Het eenvoudigste is de trein via een relais aan en uit te zetten. We gebruiken dan maar één bitje van de I/O poort. Dit vind ik persoonlijk geen mooie manier van besturing. Een voordeel is wel dat het voor ieder type trein werkt.

Een mooiere oplossing is onderstaande schakeling. Er worden hier vier bits gebruikt. De vier weerstanden vormen een eenvoudige DA omzetter. We kunnen dan de trein in 16 stappen laten versnellen en vertragen. Dit is echter alleen bruikbaar voor treintjes die op gelijkstroom rijden. Machinisten die dieseltreinen en treinen op wisselstroom besturen zullen dus wat anders moeten verzinnen.



Figuur 1 : DAC trein snelheidsbesturing

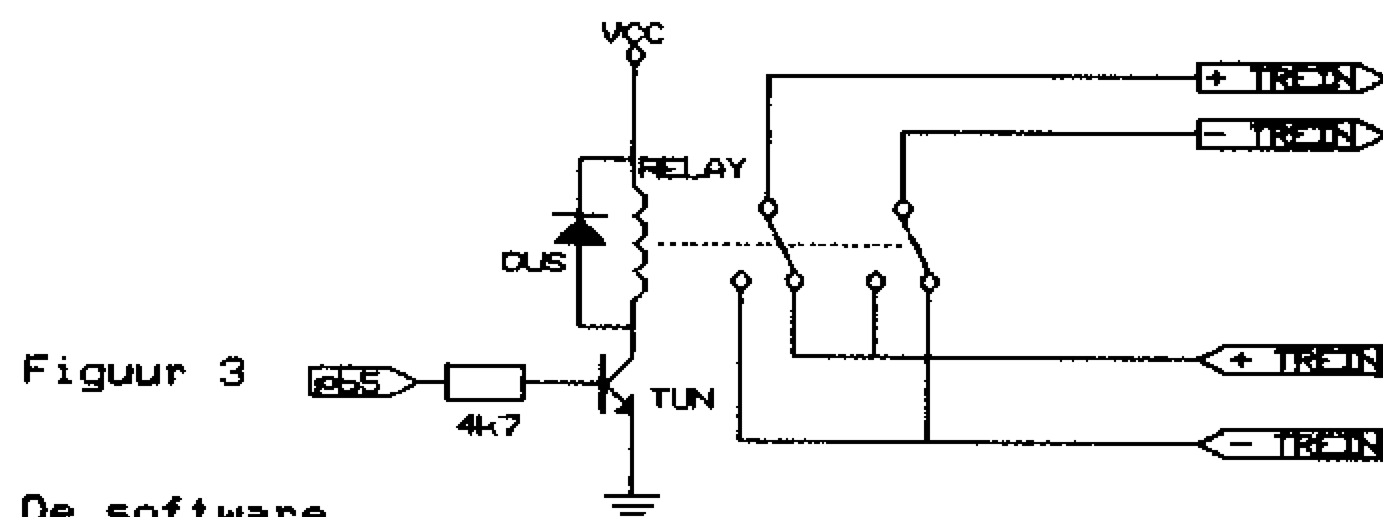
Enkele decimeters voor het station heb ik een lichtsluis geplaatst. Als de trein deze lichtsluis voorbij rijdt kan er een interrupt gegeven worden (voor gevorderde programmeurs) of er wordt steeds gekeken of er een trein voorbij rijdt (minder gevorderde programmeurs).



Figuur 2 : Lichtsluis

Als u een acht bits I/O poort gebruikt, bijvoorbeeld de B-poort van de VIA dan blijven er nog ongeveer drie bits over die we kunnen gebruiken om de verlichting aan en uit te schakelen op de modelbaan. Dit kan eenvoudig m.b.v. een relais. Als we een fotocel toepassen kan de computer geheel automatisch het licht aan maken als het in uw kamer donker wordt. Dat laat ik geheel aan uw eigen fantasie over.

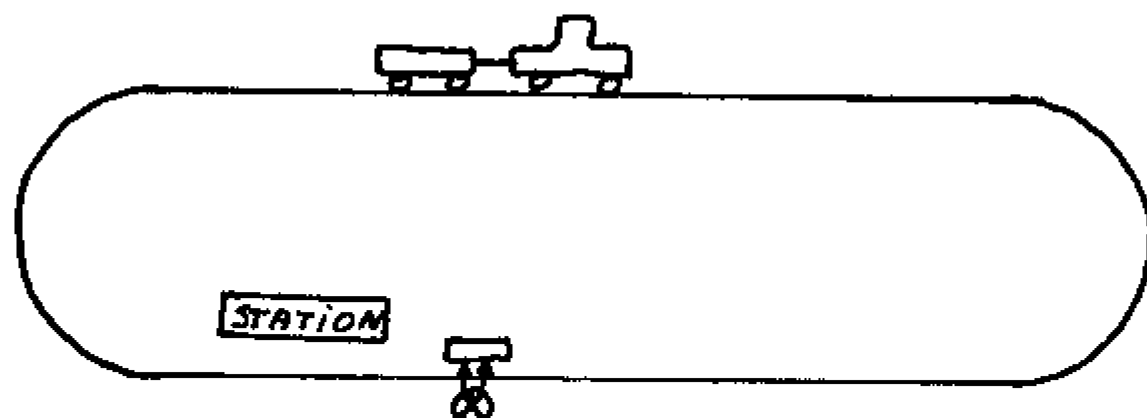
Reserveer ook een bit voor het omkeren van de rij-richting van de trein. Dat kan met onderstaande schakeling:



Figuur 3

De software

Een algemeen programma is meestal moeilijk te geven omdat alle modelbanen verschillend zijn. Daarom geef ik u enige richtlijnen om zelf zo'n programma te schrijven.



Figuur 4 : een schets van de modelbaan

U begint het programma met het resetten van het systeem. Dit kunt u doen door de trein net zolang te laten rijden totdat de lichtsluis gepasseerd wordt. De trein wordt dan tot stilstand gebracht bij het station.

Vervolgens wacht de computer een tijdje en laat dan de trein een rondje rijden en weer stoppen. Na een aantal rondjes worden de lampjes aan of uit gemaakt. Deze cyclus kan zich dan weer herhalen. De trein rijdt dan geheel automatisch.

Dit is best aardig, maar u kunt natuurlijk ook de mogelijkheid inbouwen dat u vanaf het toetsenbord de trein bedient. Bijna alles is mogelijk, als u maar programmeert!

Enkele standaard procedures

Deze procedures maken het programma overzichtelijker en eenvoudiger. Ik ga uit van de volgende aansluitingen:

- snelheidsregeling op B-poort bits 0...3
- richting omkeren op B-poort bit 4
- verlichting op B-poort bit 5
- lichtsluis op B-poort bit 7 (input)

```
PROC INIT
  ?#B802=#7F              ; REM BIT 7 INPUT, REST OUTPUT
  VERSNELTOT(10)          ; REM LAAT TREIN RIJDEN
  DO UNTIL SLUIS          ; REM WACHT TOT SLUIS GEPASSEERD IS
  VERTRAAGTOT(0)          ; REM STOP DE TREIN
PEND
```

Onderstaande procedures versnellen/vertragen de trein totdat de uitgang de opgegeven waarde bevat. Opmerking: S is eindsnelheid van procedure, H is huidige snelheid.

```
PROC VERSNELTOT(S),H
  H=?#B800 & #0F
  WHILE H < S
    ?#B800=?#B800+1
    H=H+1
  WEND
PEND


PROC VERTRAAGTOT(S),H
  H=?#B800 & #0F
  WHILE H > S
    ?#B800=?#B800-1
    H=H-1
  WEND
PEND
```

Onderstaande procedures schakelen de rij-richting om en de verlichting.

```
PROC RECHTSOM
  ?#B800=?#B800 ! #10
PEND

PROC REVERSE
  ?#B800=?#B800 : #10
PEND

PROC LINKSOM
  ?#B800=?#B800 & #EF
PEND

PROC LICHTAAN
  ?#B800=?#B800 ! #20
PEND

PROC KNIPPER
  ?#B800=?#B800 : #20
PEND

PROC LICHTUIT
  ?#B800=?#B800 & #DF
PEND
```

Tot slot nog even een functie die de waarde TRUE afgeeft als de trein de lichtsluis passeert.

```
FUNCTION SLUIS
  SLUIS = ?#B800 & #80
FEND
```

Het programma om de trein automatisch rondjes te laten rijden kan dus als volgt uitzien:

```
PROGRAM TREINTJE
        .
        .
PROCEDURES & FUNCTIES
        .
        .
INIT
DO
  VERSNELTOT(15)
  DO
    WAIT
  UNTIL SLUIS
  VERTRAAGTOT(0)
  KNIPPER
  PAUSE 180
UNTIL FALSE
```

Dit is dus tamelijk eenvoudig. Misschien is enige aanpassing voor uw eigen systeem noodzakelijk. U heeft nu enkele richtlijnen waar u vanuit kunt gaan.

Misschien wilt u meer, en dat kan natuurlijk. In onze club is er nog geen groep die zich specifiek bezig houdt met treintjes, maar in ons land zijn wel van dergelijke verenigingen. Kijk maar eens bij HCC, die hebben wel een gebruikersgroep voor modelbaan liefhebbers. Hun adres vindt u in iedere HCC nieuwsbrief. (Hiervan is altijd wel een exemplaar aanwezig op onze regio-avond in Sittard.)

Hier rond ik de modelbaan sturing af. Ik hoop dat u er iets aan heeft. Verder wens ik u nog veel plezier met uw treintje.

Met de vriendelijke groeten van Roland.

Gokkast-2

POWER -SPIN

Gemaakt door: Ger Op Het Veld
Boxen: Geen
Geheugen: #2800-#9FFF
--*-*-*-*-*

GOKKAST-2
Een nieuwe fruitmachine.
--*-*-*-*-*

Na "Gokkast" komt nu "Gokkast-2", door Ger op het Veld

Evenals Gokkast heeft Gokkast-2 twee beeldschermen. En Gokkast-2 heeft evenals zijn voorganger geen toolboxen nodig.
De vorige Gokkast gebruikte geheugen van #1000 tot #A000. Gokkast-2 gebruikt geheugen van #2800 tot #A000. Er is nu ook een versie van de vorige Gokkast te krijgen, die maar geheugen nodig heeft van #2900 tot #A000.
Men kan het programma opstarten door het basic hoofdprogramma "Gkkst-2" in te laden en dit te runnen. De andere files worden dan ingeladen.
Het programma Gokkast-2 bestaat uit 11 files.

Nu volgt een korte beschijving van de files:

GKKST-2 :Dit is het basic hoofdprogramma van #2900 tot #4F00.
2MACH58 :Is de machinetaal van #5800 tot #6500.
- Draaien van de rollen van #5800 tot #5D00.
- Knipperen van de lampjes en geluid van #5D00 tot #6000
- De shapes zitten van #6000 tot #6500.

2BASC28 :Dit is een basic hulpfile van #2800 tot #2900
2BASC65 : " " " " " " #6500 tot #6800
2BASC98 : " " " " " " #9800 tot #A000
VDU :Is een programma om cijfers en letters te printen in clear4 mode van #5000 tot #5800.
2BONS68 :Is het tweede beeldscherm van #6800 tot #8000. Beeldscherm Bonus.
2POWER :Dit is een afbeelding van de naam van Gokkast-2, de naam is "POWER-SPIN".
2CONTRL :Is een afbeelding waarop de bediening van de toetsen staat beschreven.
2PUNTEN :Dit is het scherm waar de waarde van de plaatjes staat aangegeven.
2SCHM80 :Is het hoofdbeeldscherm van #8000 tot #9800.

De werking van het spel:
========================

GOKKEN:
~~~~~~~

Bij deze Gokkast is de ONDERSTE LIJN de winlijn. Dit is even wennen. Ook heeft hij maar drie rollen. Als men twee de zelfde plaatjes heeft op de twee meest linkse rollen en op de onderste lijn. Dan kan men de punten pakken, of gokken voor meer punten.
Natuurlijk als men drie de zelfde plaatjes naast elkaar heeft op de winlijn kan men ook gokken of de punten pakken.
Gokken is de 'Spatie balk' en punten nemen is toets 'T'.
~~~~~~~

Als men wil gokken dan kan men kiezen tussen '1' of '0'. Deze waarde '1' of '0' is de waarde van het absolute randomgetal '2'.
Men kan het beste een of twee keer '1' gokken en dan op '0' gokken. Dus afwisselend.

DE PIJLEN:
~~~~~~~~~~

Af en toe zitten er pijlen in de plaatjes. In de linkse rol wijzen de pijlen naar links. In de rechtse rol wijzen de pijlen naar rechts. En in de middelste rol wijzen de pijlen naar links en rechts.
Deze pijlen besturen de plaats van de 'AS' , bovenaan in het beeld van het hoofdbeeldscherm. (De positie wordt aangegeven door het oplichten van de vakken.)
Als in de middelste rol een pijl valt op de winlijn dan kan men een keuze maken uit de twee ernaast liggende vakken. (Links of rechts.) Wanneer het vak oplicht dat men wil hebben dan moet men op de 'Spatie balk' drukken.
De 'AS' bestaat uit zeven vakken, vier vakken waar men punten kan winnen (Bank, Nudge, 7 Bonus en Bar Bonus) en uit drie tussenvakken. Bij de tussenvakken kan men geen punten winnen.
Als er drie pijlen vallen op de winlijn dan kan men een keuze maken uit alle vakken. (Bank, Nudge, 7 Bonus en Bar Bonus.)

BANK:
~~~~~

De 'Bank' wordt gevuld door de plaatjes van de kersen waar een '10' in staat. Als de 'Bank' goed gevuld is, kan men deze leeg halen. Dit moet gebeuren als volgt:
Als het vak van de 'Bank' oplicht en als men twee dezelfde plaatjes links of drie dezelfde op de winlijn heeft. Dan moet men gokken en dan moet men direct de eerste keer mis gokken.
Dan worden de punten van de 'Bank' bij de credits geteld.
Als men goed gokt of men pakt de punten dan blijven de punten in de 'Bank' staan.

NUDGE:
~~~~~~

Als men nudge heeft dan kan men de plaatjes van de rollen laten zakken. (Van boven naar beneden) Hoeveel nudge dat men heeft wordt aangegeven onder in het vak 'Nudge' in de 'As' van het hoofdbeeldscherm. Als men begint met het spel staat daar een '0'. Nudge kan men op de volgende manier krijgen:
Het vak 'Nudge' moet oplichten in de 'As' van het hoofdbeeldscherm. Dan moeten er op de winlijn GV's vallen. Als er een of meerdere GV's vallen dan verschijnt er op de monitor het tweede beeldscherm. (Beeldscherm Bonus)
Hier gaat het vak 'Nudge Bonus' oplichten. Daarboven zitten drie rijen met vier vakken. (Er staat GV in)
Van deze drie rijen gaat om de beurt een lampje aan. (Heel snel) Als men nu op de 'Spatie balk' drukt dan blijft het lampje wat net aan ging continu branden. Dit is een reactie spel.
Als nu alle lampjes van een rij aan zijn, dan krijgt men 'Nudge'. (Zoveel als er boven de bepaalde rij staat aangegeven.)
~~~~~~

Deze nudge kan men ten alle tijden gebruiken. (Alleen niet na dat men de rollen heeft vast gehouden.) Ze hoeven niet direct op nadat ze gewonnen zijn.
De computer maakt u er op attent wanneer op de twee linkse rollen, twee de zelfde plaatjes zitten. Twee dezelfde plaatjes zijn meestal niet veel punten. Men moet dan zelf kijken of op de derde rol (de rechtse) ook nog zo'n plaatje zit. Dan is het de moeite waard om nudge te pakken.

7 BONUS:

Als het vak '7 Bonus' oplicht in de 'As' van het hoofdbeeldscherm dan moeten er zoveel mogelijk zeven's vallen. (Ze kunnen op de winlijn vallen maar ook op de bovenste rij en de middelste rij.)
Als er een zeven valt dan gaat de computer naar het beeldscherm bonus. Hier licht het vak '7 Bonus' op. Onder het vak '7 Bonus' van het beelscherm bonus zijn negen vakken waarin een 7 staat. Nugaat het vak waarin een 7 staat oplichten wat de zelfde positie heeft als de 7 in de rollen van het hoofdbeeldscherm. Als een rij met zeven's brandt. (Drie vertikaal of drie horizontaal.)Dan krijgt men punten ervoor. Hoe meer rijen met zeven's dat aan zijn, hoe meer punten. Als men b.v. 20 punten heeft kan men deze nemen (Toets '7') en dan bij het hoofdbeeldscherm gokken voor meer punten. Men kan ook de zeven's sparen tot men maximaal 200 punten heeft.
MAAR als het vak '7 Bonus' oplicht in de 'As' van het hoofdbeeldscherm en als dan twee of drie dezelfde plaatjes op de winlijn vallen dan moet men direct de eerste keer mis gokken, anders gaan de vakken met zeven's die oplichten in het beeldscherm bonus uit.

BAR BONUS:

Als het vak 'Bar Bonus' oplicht in de 'As' van het hoofdbeeldscherm. Dan moeten er zoveel mogelijk Bar's vallen op de winlijn. Telkens als er een Bar valt dan gaat de computer naar het beeldscherm bonus. Hier gaat het vak 'Bar Bonus' branden. Nu volgt weer een reactiespel. Hetzelfde als bij 'Nudge'. Men kan hier 50, 100 of 200 punten winnen. Als men bij 'Bar Bonus' punten heeft gewonnen dan kan men terug bij het hoofdbeeldscherm gokken voor meer punten.

De speelwijze van het spel:

Als men wil spelen met deze fruitmachine, moet men minstens een half uur tiid er voor nemen. De eerste minuten heeft men weinig kans om veel punten te winnen.
Men moet in het begin rond de 10 nudge bijeen verzamelen. Dan kan men bij '7 Bonus' en 'Bar Bonus' punten gaan verzamelen. Ondertussen moet men goed opletten of er nudge mogelijk is. De computer geeft dit wel aan maar niet altijd. Als men een tijdje heeft gespeeld kan men de 'Bank' gaan leeghalen. Dit kan men het beste doen met een paar nudge. Namelijk:

= Het vak 'Bank' moet oplichten.
- Als nudge mogelijk is moet men deze pakken.
- Dan de eerste keer mis gokken.

Men kan ook tijdens het spel de stand van het beeldscherm bonus bekijken. Men moet dan toets 'S' indrukken. (Swap)

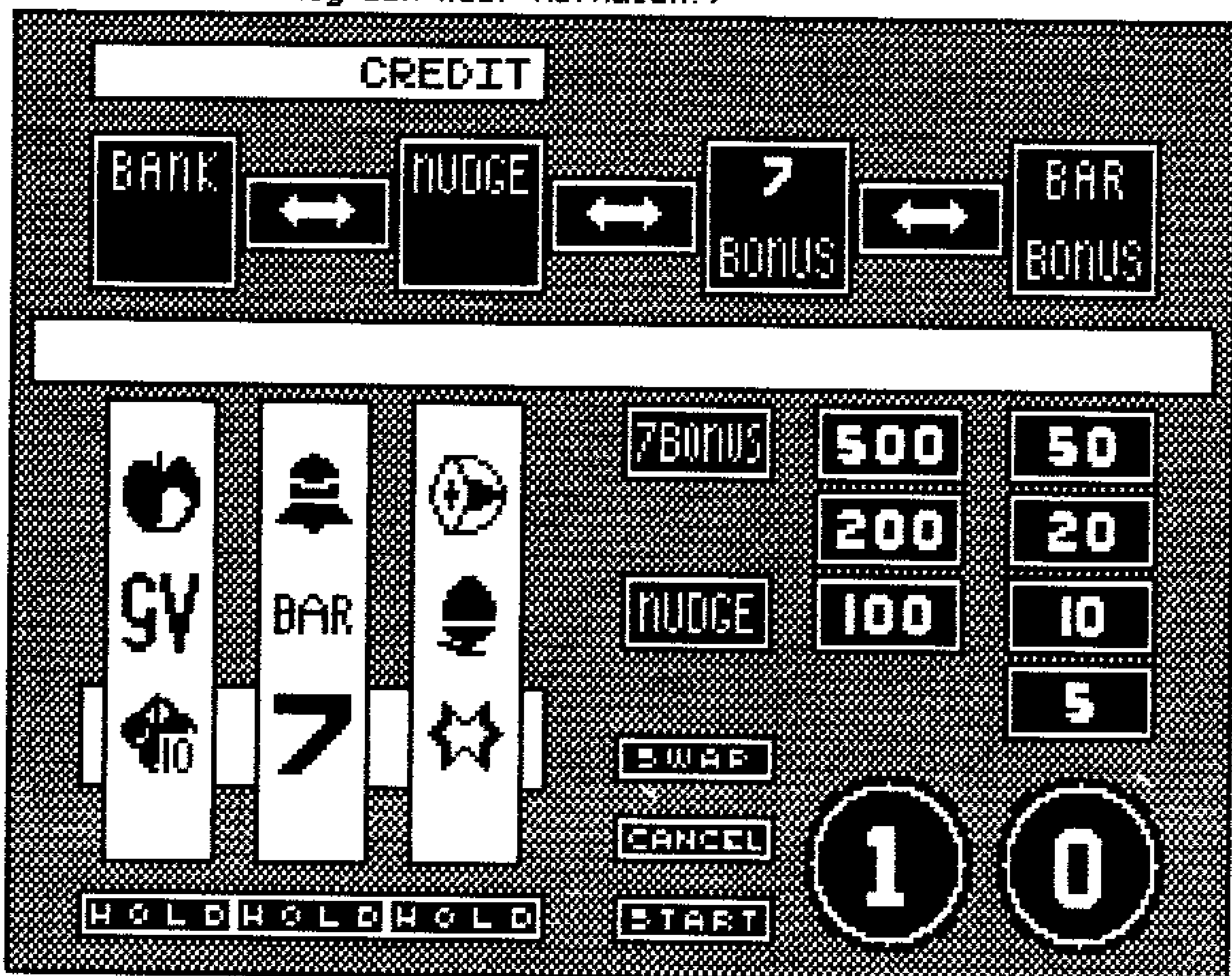
Verder kan men b.v. GV's en Bar's vast houden of plaatjes met pijlen erin. Net zoals het uitkomt. Deze gokkast heeft geen automatische 'Hold'.

In het hoofdbeeldscherm en in het beeldscherm bonus is een wit vak waarin af en toe teksten verschijnen. Hierin staat de bediening van de toetsen beschreven.

De bediening van de toetsen:

Start (Draaien rollen)	= 'Spatie balk'
Hold (Vast houden rollen)	= Toets '1,2 en 3'
Cancel (Los maken rollen)	= Toets 'C'
Swap (Beeldscherm verwisselen)	= Toets 'S'
Gokken	= 'Spatie balk'
Punten nemen	= Toets 'T'
Keuze pijlen (In de 'As')	= 'Spatie balk'
Reactie spel 'Bar Bonus'	= 'Spatie balk'
Reactie spel 'Nudge Bonus'	= 'Spatie balk'
Punten pakken '7 Bonus'	= Toets '7'

Nudge = Eerst toets 'N' in drukken en dan de plaatjes laten zakken door toets '1,2 of 3' in te drukken. (Als men een plaatje twee plaatsen wil laten zakken moet men dit nog een keer herhalen.)



Wilt u lid worden van de ATOM COMPUTER CLUB?
Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Deze kan u inlichten omtrent het lidmaatschap.

Regio TWENTE;
G.J.Noorland Prinses Ireneweg 4 7433 DE Schalkhaar
05700-25294

Regio NOORD-HOLLAND;
P. van Kuik Zuideinde 54-a 1843 JP Groot-Schermer
02997-1902

Regio DEN HAAG;
Th.Waayer L.Couperusstraat 6 2274 XP Voorburg
070-862504

Regio ARNHEM;
J.Hartog Keyenbergseweg 60 6871 WK Renkum
08373-13757

Regio ZEELAND;
E.Gijssel Ruysdaelstraat 6 4462 AD Goes
01100-32557

Regio BRABANT-OOST;
J.Teulings K. Doormanstraat 54 5224 GL Den Bosch
073-212888

Regio LIMBURG;
J.Colen Provincialeweg Z-27 6438 BA Oirsbeek
04492-1957

Regio BELGIE;
Zie Regio Limburg

Leden, die behoren tot opgeheven regio's, danwel regio's die conform de statuten geen lid meer zijn van de federatie, worden in verband met de financ. administratie en verzending van ATOM-NIEUWS, door de federatie toegewezen aan nabije regio's.
Zo men tegen de regio van indeling, om welke reden dan ook, bezwaar heeft, kan men de regio van eigen keuze opgeven aan de penningmeester van de Federatie: T.Rutten, zie pag. 2 van dit blad.

Bij het aangaan van het lidmaatschap kunt u de contributie overmaken op de rekening van de federatie.Vermeld hierbij uw volledige naam, adres en de regio waar u bij ingedeeld wilt worden.